AGLM: A Hybrid Modeling Method of GLM and Data Science Techniques

Suguru Fujita* Toyoto Tanaka† Kenji Kondo‡ Hirokazu Iwasawa§

2020-03-15

Abstract

In recent years, one of the most critical tasks for actuaries is to adopt data science techniques in predictive modeling practice. However, due to the peculiarity of insurance data as well as the priorities taken by actuaries in decision-making, such as the interpretability of models and regulatory requirements, most actuaries may find difficulties in applying them. We believe some original modeling methods with a good balance of high predictive accuracy and strong explanatory power is what is required. We propose, from this standpoint, AGLM (Accurate GLM), a simple modeling method with a desirable good balance accomplished by combining data science techniques and conventional Generalized Linear Models. For practitioners' convenience, we have also developed an R package named `aglm` (https://github.com/kkondo1981/aglm). Since the first version released in January 2019, the `aglm` can make numeric features segmented optimally exactly as Fused LASSO does when the L1 regularization is designated. In addition, the current version can, alternatively if preferable, change them from linear variables to the optimal piece-wise linear variables. Those functions make the constructed predictive model much more flexible than a conventional GLM hopefully still keeping sufficient explanatory power.

1 Introduction

In recent years, it has become an essential task for actuaries to integrate machine learning and data science techniques into actuarial practices as those techniques have been developing remarkably. There are many types of recent research on the implementation of machine learning methods, including the gradient boosting machine (GBM) and the neural network (NN), for pricing, reserving evaluation, and so on (Yang, Qian, and Zou 2018), (Poon and others 2019). These modeling methods achieve high prediction accuracy by specifying the important features among many candidates and fully capturing the complicated non-linear relationship between the features and the response variable. These methodologies will be indispensable to evaluate risks precisely for actuaries who are working in the information society, where a massive amount of data is being produced every second.

On the other hand, actuaries have traditionally developed predictive modeling by themselves to forecast future events from the given data (Frees, Derrig, and Meyers 2014). In this area, they have chosen to use the generalized linear model (GLM), the generalized additive model (GAM), and so on, which can clearly model relationships between the features and the response variable, and hence are interpretable. It is essential to use interpretable models with the knowledge of

*Guy Carpenter Japan, Inc., suguru.fujita@guycarp.com
†Tokio Marine & Nichido Fire Insurance Co., Ltd., toyoto.tanaka@tmnf.jp
‡Tokio Marine & Nichido Life Insurance Co., Ltd., kenji.kondou@tmn-anshin.co.jp
§iwahiro@bb.mbn.or.jp

insurance business used in actuarial practices such as claim analysis, pricing, reserving evaluation, and cash flow prediction. For instance, actuaries have accountability to their stakeholders (such as regulators, auditors, etc.), and they are expected to ensure that models are reflecting the knowledge shared among insurance practitioners and then determine the distribution of the response variable.

Given the situation above, it seems useful to create models with high interpretability and high prediction accuracy. It is naturally not easy, as there is a trade-off between high interpretability and high prediction accuracy, but is likely becoming a trend in both machine learning and predictive modeling areas. For example, pieces of research on the interpretability for AI and machine learning have been getting attention recently (Gunning 2017). Also, in predictive modeling, initiatives on balancing between prediction accuracy and interpretability have been increasing. For instance, Devriendt et al. (2018) proposed the regularized GLM with different types of regularization term according to the type of features such as integer, ordinal, categorical, aiming to enhance interpretability. While Wüthrich and Merz (2019) suggested a method combining GLM and NN to improve the predictive accuracy of GLM, which can be interpreted as a way to hire high interpretability of GLM and high predictive accuracy of NN simultaneously. In the future, we expect that the initiatives on well-balanced models between high predictive accuracy and high interpretability will become more and more critical.

Our model, Accurate Generalized Linear Model (AGLM), is developed to achieve both high interpretability and high predictive accuracy. It is based on GLM and equipped with recent data science techniques. High interpretability and high predictive accuracy are achieved at the same time in the following ways.

- AGLM has a clear one-to-one relationship between the features and the response variable, as it is based on GLM. Therefore, it is possible to explain clearly whether the increase in a certain variable always gives a negative or positive effect on the response variable when the other features remain the same. Machine learning methods like GBM and NN often do not have this property. Users can usually explain only average relationships between the features and the response variable, but it is not always the case. However, this could cause issues when we explain about features of interest, for example, differences of output according to the gender and/or age of each insured.
- AGLM can avoid both underfitting and overfitting even when there are many features and/or highly complicated relationships between the features and the response variable by using data science techniques. High predictive accuracy is be achieved by introducing discretization, two specific transformations of features (we call them O dummy variables and L variables respectively), and regularization into GLM. We will show the details later. As a result, our numerical experiments show the high prediction accuracy of the AGLM, which is comparable with GBM.

Note that it is important for us to provide an environment where many actuaries can easily use AGLM because it aims to satisfy the required properties for actuarial practices. Therefore, we developed an R package named `aglm` for AGLM and shared it as an open-source. Hence, the numerical experiments by this package demonstrates how to apply it to insurance data.

Our paper consists of the following chapters. In Chapter 2, we explain GLM and the regularized GLM, which are base models for AGLM. In Chapter 3, we define AGLM and describe techniques used in AGLM. And, we provide the outline of `aglm` package. In Chapter 4, we qualitatively compare AGLM with other existing models in terms of both interpretability and prediction accuracy. In Chapter 5, we quantitatively evaluate AGLM through examples of Poisson regression for insurance data. Finally, in Chapter 6, we conclude.

2 GLM and Regularized GLM

In this chapter, we consider GLM and the regularized GLM, which are the base models for our AGLM.

GLM (Nelder and Wedderburn 1972) is a model that constructs the following relationship between the expected value of each response variable and the features, assuming that the error distribution of a response variable belongs to the exponential distribution family.

$$E[y_i] = g^{-1}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \quad (i = 1, 2, ..., n), \tag{1}$$

where n is the number of observations, y_i is i-th response variable, x_{ij} $(j = 1, 2, ..., p)$ are i-th features (p is the number of features), and $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)^T$ represents regression coefficients. g is called the link function that is differentiable and strictly monotonic and links the expected value of y_i with the linear combination of features $\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$. The estimated value $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is given by maximum likelihood estimation (MLE) as the solution of the following minimization problem.

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{-\log L(\boldsymbol{\beta})\}, \tag{2}$$

where $L(\boldsymbol{\beta})$ is the likelihood function.

Then, the regularized GLM is a model that applies regularization to GLM (Friedman, Hastie, and Tibshirani 2010). The regularization is a method that penalizes the complexity of the model by adding the regularization term to the objective function. Specifically, the estimation of $\boldsymbol{\beta}$ for the regularized GLM can be done by extended maximum likelihood estimation as follows, where $R(\boldsymbol{\beta}; \lambda)$ is the regularization term.

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\{-\log L(\boldsymbol{\beta}) + R(\boldsymbol{\beta}; \lambda)\}. \tag{3}$$

As typical regularization terms, Ridge (Hoerl and Kennard 1970), Lasso (Tibshirani 1996), and Elastic Net (Zou and Hastie 2005) are widely used. Note that the intercept is not generally included in the regularization, and thus the subscript j ranges from 1 to p.

$$\begin{aligned}
&\text{Ridge (L2 regularization)} \quad R(\boldsymbol{\beta}; \lambda) = \lambda \sum_{j=1}^{p} |\beta_j|^2, \\
&\text{Lasso (L1 regularization)} \quad R(\boldsymbol{\beta}; \lambda) = \lambda \sum_{j=1}^{p} |\beta_j|, \\
&\text{ElasticNet} \quad R(\boldsymbol{\beta}; \lambda, \alpha) = \lambda \left\{ (1-\alpha) \sum_{j=1}^{p} |\beta_j|^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right\}.
\end{aligned} \tag{4}$$

Here, λ, α $(\lambda > 0,\ 0 \leq \alpha \leq 1)$ are hyperparameters given by users, which determine the regularization effect. They are generally determined by the holdout or the cross validation (CV). The Elastic Net includes Ridge ($\alpha = 0$), Lasso ($\alpha = 1$) and mixed model ($0 < \alpha < 1$) as its special cases. The effects of regularization is as follows, depending on the types of regularization terms.

- Avoid the instability of calculation result due to multicollinearity.
- Deal with ill-conditioned problems ($p > n$).
- Achieve feature selection (automatically select the effective features among a lot of candidates).

- Avoid the decline in prediction accuracy due to overfitting.

The regularization is widely used in the context of machine learning and data science. It might be mainly because the regularization solves the difficulties in the modeling of high dimensional data that many people face in these areas. Note that only the regularization including L1 norm has the feature selection effect (*i.e.* Lasso and Elastic Net with $0 < \alpha < 1$).

You can easily use the regularized GLM via R package `glmnet` (Jerome Friedman and Qian 2019), for example.

3 AGLM: Accurate Generalized Linear Model

In this chapter, we describe our proposed method AGLM and its components.

3.1 Definition of AGLM

AGLM[1] is defined as a regularized GLM which applying a sort of feature transformations using a discretization of numerical features and specific coding methodologies of dummy variables. We explain the details of AGLM components and then formulate AGLM.

3.2 The discretization of numerical features

The discretization is a very simple and widely used method, splitting a numerical feature into several bins as groups. Here, we focus on the discretization of numerical features. Let us introduce a numerical feature x defined on $(b_0, b_m]$, m bins by $B_1 = (b_0, b_1], B_2 = (b_1, b_2], ..., B_m = (b_{m-1}, b_m]$ for x, and the contribution β_j of each bin B_j to the response variable y as a constant. The discretization of numerical features enables actuaries to reflect non-linear effects into models. It is actually possible to approximate any non-linear function by the contribution curve (step function form) $\sum_j \beta_j \mathbb{I}_{B_j}(x)$ where $\mathbb{I}$ is an indicator function, so the underfitting is effectively avoided compared to the case where the feature are used as is. Note that we also consider alternative discretization not using step functions in AGLM as described later.

It is important to decide the number of bins in the discretization. Coarse bins have difficulties in avoiding underfitting, while too small bins might cause overfitting. Although in traditional way, the number or width of bins has to be determined manually, with actuarial expertise, by considering the nature of features, the regularized GLM can automatically determine it using the regularization. In other words, we can expect to avoid overfitting automatically by just dividing each numerical features as much as possible and letting the regularization (Lasso or Elastic Net with $0 < \alpha < 1$, having feature selection effects) judge whether the contribution β_j of each bin should be included ($\beta_j \neq 0$) or excluded ($\beta_j = 0$) in the model.

3.3 Coding of numerical features with dummy variables

In this section, we describe how AGLM codes numerical features with special dummy variables. As mentioned above, the combination of the discretization and the regularization solves the problem of both underfitting and overfitting. However, we should consider the ordinal information[2] between each bin in the discretization of numerical features. If each bin's coefficient is estimated independently, the entire component curve could lack consistency (like in the case where $\beta_{j-1} \neq 0, \beta_{j+1} \neq 0$, but $\beta_j = 0$) or the coefficient would change sharply among the adjacent bins,

[1] AGLM is named as "Accurate" GLM because it can be expected to achieve higher prediction accuracy than usual GLM, but many words that express the characteristics of AGLM such as "Actuarial," "Accountable," etc., happen to begin with the same letter 'A.' Therefore, we see "A" in "AGLM" as a somewhat symbolic letter representing all of these words.

[2] It means order ranking or magnitude relationship. Numerical features and ordinal features have that information.

resulting in the non-smoothness of contribution curve. For example, in the premium calculation of insurance policies, if the claim frequency model were discretized by age band, it might cause an output that some specific age groups can receive discounts on insurance premiums while the nearby groups cannot. To avoid this happens, AGLM implements a feature coding using special dummy variables.

3.3.1 O dummy variables

Firstly, we define usual dummy variables. When the discretized feature x takes m levels $\{1, 2, ..., m\}$, the dummy variables $d_1(x), d_2(x), ..., d_m(x)$ are defined as follows.

$$d_j(x) = \begin{cases} 1, & \text{if } x = j \quad (j = 1, 2, ..., m); \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Regression analysis of categorical features can be performed by coding them using dummy variables. This means that the contribution of features x to the response variable y can be expressed by a linear combination $\sum_j \beta_j d_j(x)$, where each regression coefficient β_j (corresponding to each dummy variable $d_j(x)$) is estimated independently and the relationship between each level is not reflected.

On the other hand, as mentioned above, there actually exist ordinal relationships between each level (corresponding to each number of the bin) for the discretized numerical features. Therefore, we introduce the following dummy variables $d_1^O(x), d_2^O(x), ..., d_m^O(x)$ to capture it.

$$d_j^O(x) = \begin{cases} 1, & \text{if } x < j \quad (j = 1, 2, ..., m); \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

In other words, $d_j^O(x)$ would be 1 when the level is less than j and 0 otherwise. This kind of coding is called split coding (J. Gertheiss and Tutz 2009) or thermometer encoding (Garavaglia and Sharma 1998) depending on the context. In the AGLM, we call it O (meaning "Ordinal") dummy variables to clarify our purpose of reflecting the ordinal relationships, while we call usual dummy variables (Equation (5)) U dummy variables hereafter.

Let us see the difference between the U dummy variables and the O dummy variables. Table 1 and 2 shows the dummy matrices of U dummy variables and O dummy variables respectively when the feature is age (ranging from 10 to 89-year-old) and the levels are set every 10-year-old by discretization (level 1 for the teens, level 2 for the twenties, and the same hereinafter).

Table 1: In the case of U dummy variables

Age	x	$d_1(x)$	$d_2(x)$	$d_3(x)$	$d_4(x)$	$d_5(x)$	$d_6(x)$	$d_7(x)$	$d_8(x)$
10-19	1	1	0	0	0	0	0	0	0
20-29	2	0	1	0	0	0	0	0	0
30-39	3	0	0	1	0	0	0	0	0
40-49	4	0	0	0	1	0	0	0	0
50-59	5	0	0	0	0	1	0	0	0
60-69	6	0	0	0	0	0	1	0	0
70-79	7	0	0	0	0	0	0	1	0
80-89	8	0	0	0	0	0	0	0	1

Table 2: In the case of O dummy variables

Age	x	$d_1^O(x)$	$d_2^O(x)$	$d_3^O(x)$	$d_4^O(x)$	$d_5^O(x)$	$d_6^O(x)$	$d_7^O(x)$	$d_8^O(x)$
10-19	1	0	1	1	1	1	1	1	1
20-29	2	0	0	1	1	1	1	1	1
30-39	3	0	0	0	1	1	1	1	1
40-49	4	0	0	0	0	1	1	1	1
50-59	5	0	0	0	0	0	1	1	1
60-69	6	0	0	0	0	0	0	1	1
70-79	7	0	0	0	0	0	0	0	1
80-89	8	0	0	0	0	0	0	0	0

It is obvious that the matrix for U dummy variables is an identity matrix, while that for O dummy variables is an upper triangular matrix without diagonal elements.

Secondly, we describe how the O dummy variables code the ordinal relationship of features. In the case of regression analysis with O dummy variables, the contribution of each feature x to the response variable y can be formulated as the linear combination $\sum_j \beta_j d_j^O(x)$. Thus, β_j is added up for all j satisfying $j > x$ otherwise not. Therefore, this represents how much difference should be made between the boundary $x < j$ and $x \geq j$. Particularly, it can be expected to automatically determine whether to integrate the adjacent groups ($\beta_j = 0$) or not ($\beta_j \neq 0$) by combining the feature coding with O dummy variables and the regularization with feature selection effect (*i.e.* L1 norm). For Table 2, given that $d_3^O(x), d_6^O(x), d_8^O(x)$ are selected (which means $\beta_3 \neq 0, \beta_6 \neq 0, \beta_8 \neq 0$ and the other $\beta_j = 0$), it concludes that the ages would be divided into four groups: 10s to 20s, 30s to 50s, 60s to 70s, and 80s only, where the contribution of each group to y is constant.

It is noteworthy that the combination of the O dummy variables and L1 regularization is methematically equivalent to the Fused Lasso (Tibshirani et al. 2005), which is widely known as sparse modeling techniques. The Fused Lasso can be interpreted as adding the following terms to the loss function, having both feature selection effect (the first term) and grouping effect of adjacent features (the second term).

$$\text{Fused Lasso} \quad R(\boldsymbol{\beta}; \lambda^{(1)}, \lambda^{(2)}) = \lambda^{(1)} \sum_{j=1}^{p} |\beta_j| + \lambda^{(2)} \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|. \tag{7}$$

The combination of O dummy variables and regularization with the feature selection effect enables us to get the feature grouping effect corresponding to the second term in the Fussed Lasso (refer to the Appendix for the proof of this parity). At the same time, it achieves the same effect as the Fused Lasso. There are some advantages by using our method. For instance, we need not modify the regularization terms and can directly use L1 regularization or Elastic Net. Therefore, the formulation of the optimization problem can be simple in this sense.

Note that O dummy variables can alternatively be set to be 1 when $x > j$ instead of $x < j$, and also can be implemented to categorical features if they have orders.

3.3.2 L variables

As discussed above, the combination of the O dummy variables and the regularization enables us to reflect both non-linearity and ordinal relationships in the original data. However, the step-wise contribution curve $\sum_j \beta_j d_j^O(x)$ is not continuous and sometimes it is not desirable. AGLM has an option to implement the following ingenious coding to ensure the continuity of the contribution curve.

As in the previous discussion, let us suppose the numerical feature x is divided into m bins. Then, the L variables $l_1(x), l_2(x), ..., l_m(x)$ for x are given as follows:

$$l_j(x) = \begin{cases} |x - b_j|, & (j = 1, ..., m-1); \\ x, & (j = m). \end{cases} \tag{8}$$

We can perform regression analysis with L variables instead of x itself in the same way as with dummy variables. Then, the contribution curve $L(x) = \sum_j \beta_j l_j(x)$ is a polyline (piece-wise linear function), connecting $(m+1)$ points $(b_j, L(b_j))_{j=0}^{m}$. This is a continuous curve where the slope of each bin is constant. Note that each β_j represents the difference of slopes between the adjacent bins, and when $\beta_j = 0$, it means that the slopes of $L(x)$ in the adjacent bins are constant.

3.4 Model formulation and estimation

In the AGLM, features are transformed into multi-dimensional feature vectors by the following procedure, before they are used in regression models.

- A numerical feature x is applied discretization by binning and transformed into a new feature vector $(x, d_1^O(x), d_2^O(x), ..., d_m^O(x))$ or $(x, l_1(x), l_2(x), ..., l_m(x))$, where x is the feature itself, and $d_j^O(x)$'s and $l_j(x)$'s are O dummy variables and L variables. They way of binning x and the number of bins can be set as desired, but in our `aglm` package described later, we use equal frequency method or equal width method and set the number of bins m to 100 by default.
- A categorical feature x is, if the categories have order, transformed to a new feature vector $(d_1^O(x), d_2^O(x), ..., d_m^O(x))$, where $d_j^O(x)$'s are O dummy variables. In this case, m is equals to number of categories.[3]
- A categorical feature x without order is transformed into a new feature vector $(d_1(x), d_2(x), ..., d_m(x))$, where $d_j(x)$'s are U dummy variables.

Consider a regression problem which expresses a response variable y_i using p features $(x_{i1}, x_{i2}, ..., x_{ip})$. According to the above rule, convert each feature x_{ij} to m_j-dimensional feature vector $\boldsymbol{z}_{ij}$ and, let $\boldsymbol{\beta}_j = (\beta_{j1}, \beta_{j2}, ..., \beta_{jm_j})$ denote m_j-dimensional regression coefficients vector.

Then, AGLM is defined as a regularized GLM that formulates the expected values of a response variable according to the following relationship, where g is a link function.

$$E[y_i] = g^{-1}\left(\beta_0 + \sum_{j=1}^{p} \boldsymbol{z}_{ij}\boldsymbol{\beta}_j'\right) \quad (i = 1, 2, ..., n). \tag{9}$$

In addition, the regularization term is set to Elastic Net type as follows:

$$R(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_p; \lambda, \alpha) = \lambda \left\{ (1-\alpha) \sum_{j=1}^{p} \sum_{k=1}^{m_j} |\beta_{jk}|^2 + \alpha \sum_{j=1}^{p} \sum_{k=1}^{m_j} |\beta_{jk}| \right\}. \tag{10}$$

Note that coefficient vectors are estimated by solving the minimization problem defined as Equation (3) in Chapter 2.

[3] In the case of regression problem without regularization, it is common to avoid multicollinearity by dropping the 1st or m-th dummy variables, but since the ALGM is one of the types of regularization regression, no need to drop variables.

3.5 R package

In this section, we describe our R package named `aglm`.

We believe that it is important to invite practical actuaries to use AGLM easily. Hence, we develop an R package named `aglm`. Another merit of providing package is to encourage as many users as possible to reproduce the functions.

Note that the `aglm` package has been developed on GitHub which is a web service for open source developers so that you can see the updated source code from our GitHub page[4]. In addition, you can easily install the `aglm` package from GitHub by using `devtool` package[5].

3.5.1 Basic functions

Use `aglm` function as follows to train and fit data.

```
library(aglm) # Load aglm
fitted <- aglm(x, y, alpha = 1, lambda = 0.1)
```

Here, `x` is a data frame for the features and `y` is a vector for the response variable. The `aglm` returns a calibrated model as an R object after estimating the parameters of the AGLM which represents the relationship between `x` and `y`. The arguments `alpha` and `lambda` correspond to the hyperparameters α and λ respectively.

Note that we need to use linear terms and the O dummy (or L) variables for numeric variables, the O dummy variables for ordered categorical variables, and the U dummy variables for the others. The `aglm` function automatically determines the types of dummy variables to apply according to each feature's type in `x`. Table 3 below shows the relationship between each type of data in R and dummy variables applied by the `aglm` function. This is just a default setting and can be changed if needed.

Table 3: Feature type handling by `aglm` function

Feature type in R	Class	Ordered?	Linear term	U dummy	O dummy/L var
`numeric, integer`	Numerical	Yes	✓	-	✓
`ordered`	Categorical	Yes	-	✓	✓
`factor, logical`	Categorical	None	-	✓	-

We can then run prediction for the response variable corresponding to new features `newx` by using `predict` function with the `fitted` object described above.

```
newy <- predict(fitted, newx)
```

3.5.2 More functions

The `aglm` package provides various functions to enhance usability for users as follows (see Table 4).

Table 4: `aglm` functions

Name	Function
`aglm`	Fit AGLM
`predict`	Predict a response variable for new features

[4] https://github.com/kkondo1981/aglm
[5] use `devtools::install_github` function.

Name	Function
`plot`	Plot components or partial residuals
`cv.aglm, cva.aglm`	Determine the optimum `alpha, lambda` by CV
`coef, deviance, residuals`	Calculate regression coefficients, deviance, and residuals

For example, the relationship between each feature and a response variable can be visualized by the `plot` function. More, tuning of hyperparameter based on data can be performed by the `cv.aglm` and `cva.aglm`. The `coef, deviance, and residuals` are useful to confirm the result of analysis. Note that these functions can be used like these of `glmnet` package[6] so that users feel familiarity with `glmnet` can easily use the `aglm` package.

4 Advantages of AGLM

In this chapter, we describe what are the strong points of AGLM, comparing with other modeling methods qualitatively (quantitative evaluations will be shown in Chapter 5), and introduce our R package.

4.1 The pros of AGLM

The AGLM has been developed based on GLM, which is familiar to actuaries, resulting in high interpretability. Note that the improvement can be made by elaborating on the feature engineering with the discretization, the O dummy variables, and the L variables on the condition that GLM framework holds. Therefore, the optimization formula of AGLM falls in the range of the regularized GLM's framework. We summarize the pros of AGLM as follows.

- Hold a clear and intuitive one-to-one relationship between the features and the response variable since it is based on GLM.
- Avoid both underfitting and overfitting, even if there are a lot of features or complicated non-linear relationships between the features and the response variable. In particular:
  - Avoid underfitting by the O dummy variables and L variables for the numerical features discretized with small bins.
  - Avoid overfitting by selecting effective dummy variables through the regularization.

Furthermore, as mentioned in Chapter 3, the combination of each component of AGLM will enhance the flexibility of the model, which also contributes to enhancing the interpretability. Thus, the AGLM is a hybrid model of GLM and data science techniques, aiming to achieve a good balance between interpretability and prediction accuracy.

4.2 Advantages to other modeling methods

In this section, we qualitatively evaluate the characteristics of AGLM by comparing it with other models used in predictive modeling and data science field. We use the following models for the comparison:

- GLM (including the regularized GLM)
- GAM
- Tree-type models

4.2.1 Comparison with GLM

The GLM has been used traditionally and is a popular method in actuarial practices. The actuarial data, including claim frequency and severity, have specific properties such as taking only

[6] https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html

positive values, right-skewed distribution, and the tendency of the variance increase as the expected value rises. GLM fits these kinds of data well. It is also highly interpretable that it has a clear one-to-one relationship between the features and the response variable. AGLM also shares this nature. On the other hand, when there are many features or highly complicated relationships between the features and the response variable, GLM (except the regularized GLM) needs a manual feature selection and/or additional feature engineering such as the discretization. By going through these elaborations, GLM might achieve the equivalent prediction accuracy with AGLM. However, AGLM's automated feature engineering will achieve higher prediction accuracy more easily in many cases.

As for traditional actuarial practice, features in actuarial data such as insured age and rating class are already familiar to actuaries, and the number of them might not be so large. In this case, we do not need the regularization technique because we could do flexible modeling by inspecting data and utilizing actuarial expertise. However, it is anticipated that we will face a situation where we have to deal with a large number of features or with data which is unfamiliar to actuaries as the information society will develop further. Thus, we expect AGLM to be indispensable for actuaries as well in the future.

4.2.2 Comparison with GAM

Like GLM, GAM (Hastie 2017) has high interpretability as it has a clear one-to-one relationship between the features and the response variable. In addition, GAM can be expected to have higher prediction accuracy than GLM because its smoothing function (*i.e.* splines) captures non-linearity to some extent. On the other hand, presumably, the discretization based AGLM would work well when the relationship between the features and the response variable is non-linear but smooth, such as the case where there is a feature having a large impact in the only specific range or where the smooth function does not work well due to the inappropriateness of the explanation variable's scale.

Note that there is no clear difference between GAM and AGLM in terms of interpretability, so you can choose the method based on your preference whether you like smooth function (e.g. spline) or step function when the feature takes a constant value within each bin (or a constant slope in the case of the L variables). The smooth curve by GAM is intuitively understandable but is not always easy to deal with, and it would sometimes be inappropriate in terms of interpretability. And when making a tariff (rating table), AGLM, where each bin can have a constant value, would be rather understandable.

Finally, when there are lots of features, GAM with the regularization would effectively avoid overfitting (Chouldechova and Hastie 2015). However, there are seems to be difficulties when you try to exploit the regularized GAM in the actuarial field yet, seeing for example, the R package `gamsel` (Alexandra Chouldechova and Spinu 2018) only provides options of the normal distribution or binomial distribution.

4.2.3 Comparison with tree-type models

The tree-type models such as random forests (Breiman 2001) and GBM (Friedman 2001) are called ensemble models because they combine a large number of decision trees into a single model, which are known for the high prediction accuracy. These models can avoid both overfitting and underfitting by learning carefully with lots of parameters, resulting in capturing complicated relationships. AGLM also implements these concepts into GLM. Thus, as described in Chapter 5, the AGLM would be comparable with these methods in terms of prediction accuracy.

Furthermore, AGLM is advantageous in that there is a clear relationship between the features and the response variable. The tree-type models can also capture the relationship between the features and the response variable by the partial dependence plot (PDP), the individual conditional expectation (ICE), etc. However, these methods basically express a "marginal" relationship based

on the values averaged by the other features, only showing the tendency as a whole. For example, to explain which specific values of the policyholder attributes (30-year-old or 40-year-old, etc.) have a higher premium, the PDP will describe something ambiguousness like "We obtain this tendency as a whole but it will ultimately depend on other conditions", but AGLM will explain clearly like "We obtain this tendency decisively without exception under other conditions being the same, no matter conditions they are." Besides, when making modifications fitted models, it is clear which parameters to be changed to get desirable properties in AGLM case. This AGLM's property would be suitable when we need to explain the validity of the model considering social equality, etc., and actuaries often face to such situation. The AGLM also has an advantage that we can utilize the existing knowledge of the stakeholders to GLM and the current software etc. because of AGLM being developed by keeping GLM's framework. An `aglm` package, which is in the former section, uses the existing `glmnet` package as a back-end model so that we could significantly reduce the workload for the development.

Note that we need further discussion on how to deal with interactive effects between features if needed because we have not determined about it with AGLM, although the `aglm` package can add the interactive effects between each two-feature.

5 Numerical Experiments

In this chapter, we compare the AGLM with other modeling methods by applying to actual data and quantitatively evaluate the AGLM. And, we demonstrate how AGLM's component curves are different from those of the other methods.

5.1 Data Description

We use `freMTPL2freq` data from `CASdatasets` package of R (Charpentier 2014). This is the data of French automobile insurance and has 678,013 records with 12 features listed in Table 5. We consider `ClaimNb` as the response variable and `log(Exposure)` as offset terms. Other attributes are used as features.

Table 5: Features in `freMTPL2freq`

Feature	Description	Type	Ordered?
`IDpol`	The policy ID	Integer	Y
`ClaimNb`	The number of claims during the exposure period	Integer	N
`Exposure`	The period of exposure for a policy, in years	Real	Y
`VehPower`	The power of the car	Integer	Y
`VehAge`	The vehicle age, in years	Integer	Y
`DrivAge`	The driver age, in years	Integer	Y
`BonusMalus`	Bonus/malus, between 50 and 350: <100 means bonus, >100 means malus in France	Integer	Y
`VehBrand`	The car brand	Factor	N
`VehGas`	The car gas, Diesel or regular	Factor	N
`Area`	The density value of the city community where the car driver lives in: from "A" for rural area to "F" for urban center	Factor	N
`Density`	The density of inhabitants (number of inhabitants per square-kilometer) of the city where the car driver lives in	Real	N
`Region`	The policy region in France (based on the 1970-2015 classification)	Factor	N

Table 6 shows the number of records and sum of `Exposure` for each value of `ClaimNb`.

Table 6: Distribution of `ClaimNb`

`ClaimNb`	0	1	2	3	4	5	6	8	9	11	16
# of records	643,953	32,178	1,784	82	7	2	1	1	1	3	1
Sum of `Exposure`	336,616.1	20,670.8	1,153.4	52.8	3.1	1.1	0.3	0.4	0.1	1.1	0.3

Furthermore, we indicate density plots of numerical features and bar plots of categorical features in Fig 1 to capture the distribution of each feature. In these plots, densities are estimated by `geom_density` function of `ggplot2`, and labels of `Region` are shortened into the first two letters for visibility reason.

5.2 Experiment settings

For the experiment, we fitted five models to the `freMTPL2freq` data, namely, AGLM, GLM, Regularized GLM, GAM, and GBM. Existing R packages are used for fitting, and we list them in Table 7.

Table 7: Models and R packages used in the experiment

Model	Package
AGLM	`aglm`
GLM	`glm`
Regularized GLM (with Elastic Net penalty)	`glmnet`
GAM	`mgcv`
GBM	`gbm`

The Poisson error distribution and the log-link function are assumed for all modeling methods in common. In addition, hyperparameters for models other than GLM and other model-specific settings are chosen as follows:

- Hyperparameters α, λ of Elastic Net-type regularization are chosen by CV, using the `cva.glmnet` function of `glmnetUtils` package for the regularized GLM, the `cva.aglm` function of `aglm` package for the AGLM.
- For GAM, smoothers represented by the `s` function is applied to all the numerical features, with its default smoothing methods and parameters.
- For GBM, the optimal number of trees are chosen with CV by the `gbm` function's default method. And, parameters for optimization are set as `shrinkage = 0.01`, `train.fraction = 0.9`.
- For AGLM, L variables are used for discretized quantitative features.
- The default settings of fitting functions are applied to other settings.

Before experiments, we executed the following preprocessing to data:

- `IDpol` is discarded because it is just for identifications.
- `ClaimNb` greater than 4 are censored, judging they are outliers because policy terms of almost all the policies are one-year or less. On the other hand, `Exposure` greater than 1 are left as it is, guessing there is some data-processing reason and considering limited influences on the modeling results.
- Categorical variables are transformed into dummy variables with the default way of each fitting function, which is transforming into U dummy variables in most cases but is using either the U dummy variables or the O dummy variables properly in the AGLM case. Because

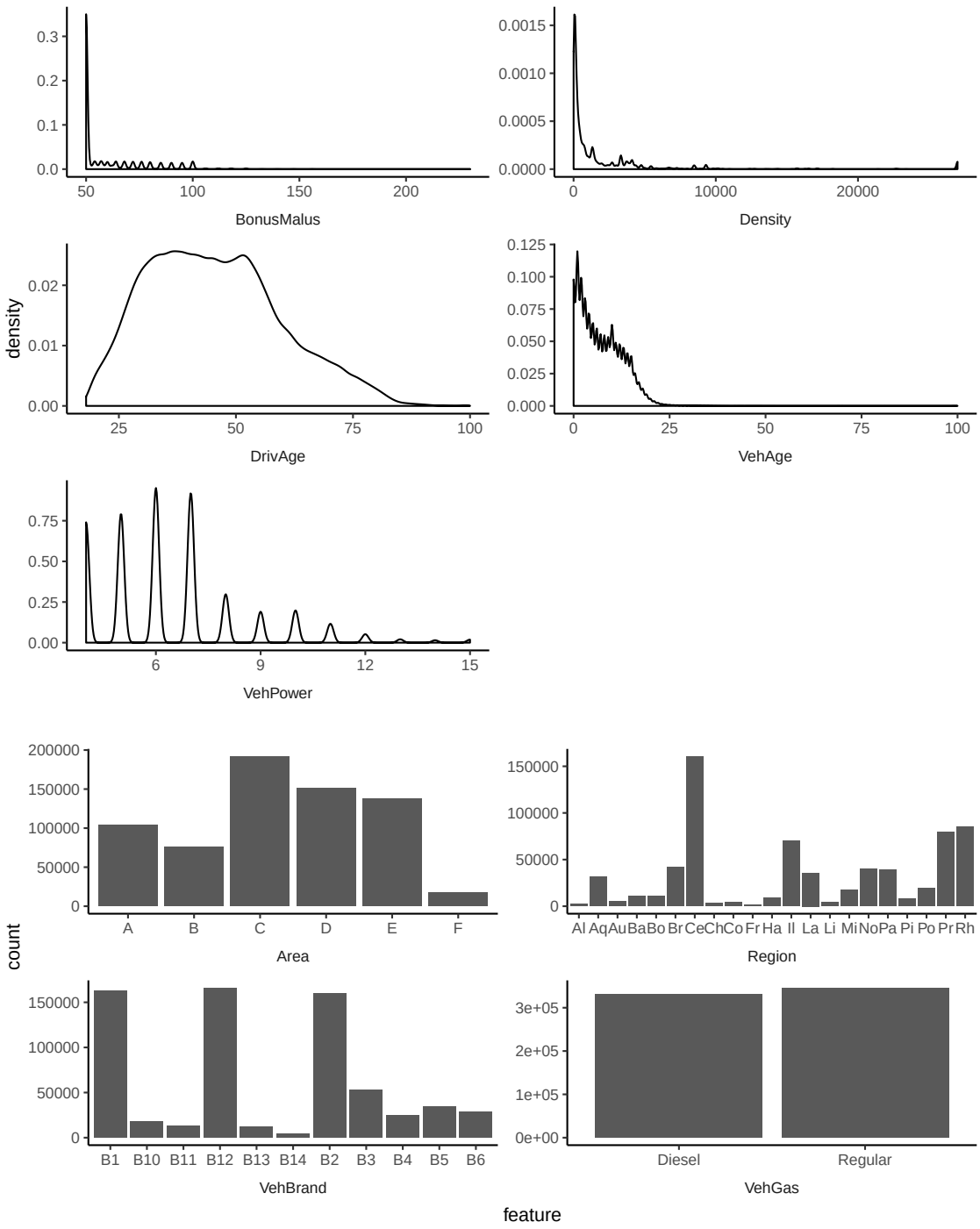

Figure 1: Distribution of features in `freMTPL2freq` (Upper: numerical features, Lower: categorical features)

the `glmnet` function cannot deal with categorical features directly, we transformed each categorical variable into the U dummy variables (without dropping neither the first nor the last dummy variable) manually.
- No feature engineering for the numerical features are applied, except those included in the modeling procedure. In practices, expert actuaries often prefer GLM with binning to model non-linearity, but only pure linear effects $\beta_j x_{ij}$ on $g(E[y_i])$ are modeled by GLM and the regularized GLM in the experiment. This is because we intended to see how the AGLM can capture the complicated effects without manual feature engineering, and these two models are placed just as baselines.

For measuring the predictive accuracies, we use holdout method. In the data-fitting phase, 75% of the entire data is randomly chosen and used as the train data for model-fitting. Then, after fitting the models, the remaining 25% of the data is used as the test data to calculate Poisson deviances as follows:

$$\frac{1}{N^{\text{test}}} \sum_{i=1}^{N^{\text{test}}} y_i^{\text{test}} \log(y_i^{\text{test}}/\hat{y}_i^{\text{test}}) - y_i^{\text{test}} + \hat{y}_i^{\text{test}}, \tag{11}$$

where N^{test} is the size of the test data, $\{\hat{y}_i\}_{i=1}^{N^{\text{test}}}$ is the fitted response variables, and $\{y_i\}_{i=1}^{N^{\text{test}}}$ is the true response variables.

5.3 Results and discussion

Now we show the result of the experiment. The deviances of the fitted models with test data are as in Table 8 (Models with smaller deviance is better).

Table 8: Poisson deviance for test data

Model	Poisson deviance
AGLM	**0.3111920**
GLM	0.3201199
Regularized GLM	0.3201245
GAM	0.3171236
GBM	0.3123919

We find that the AGLM is the most predictively accurate model for the experiment setting. The worst model is the regularized GLM, but there is almost no difference in test deviances between the regularized GLM and GLM.[7] GAM and GBM are placed between the AGLM and GLM, and GBM is better than GAM, having the second smallest deviance to the AGLM.

Next, we show how the AGLM captures the relationships between the features and the response variable. Figure 2 shows the component levels of features(*i.e.*, $\boldsymbol{z}_{ij}\boldsymbol{\beta}'_j$ in Equation (10)), corresponding to various x values. We can immediately find that component curves of the numerical features are non-linear and non-monotonic, and the complicated relationships between the features and the response variable are certainly modeled in the AGLM. Note that the way the AGLM models such relationships are purely data-driven, and in the same manner as those of modern data science techniques like decision tree-type models and NN. In fact, the PDP of the fitted GBM shows similarly complicated curves with the component curves of the AGLM. On the other hand, how the component of each feature contributes to the mean of the response variable is perfectly clear

[7] It might sound unnatural that the regularized GLM, which includes the usual GLM as a special case with $\lambda = 0$, results in the worse score than the usual GLM. This might happen because the improvement of predictive accuracy by regularization for this data is quite small and errors coming from the choice of hyperparameters are larger than it.

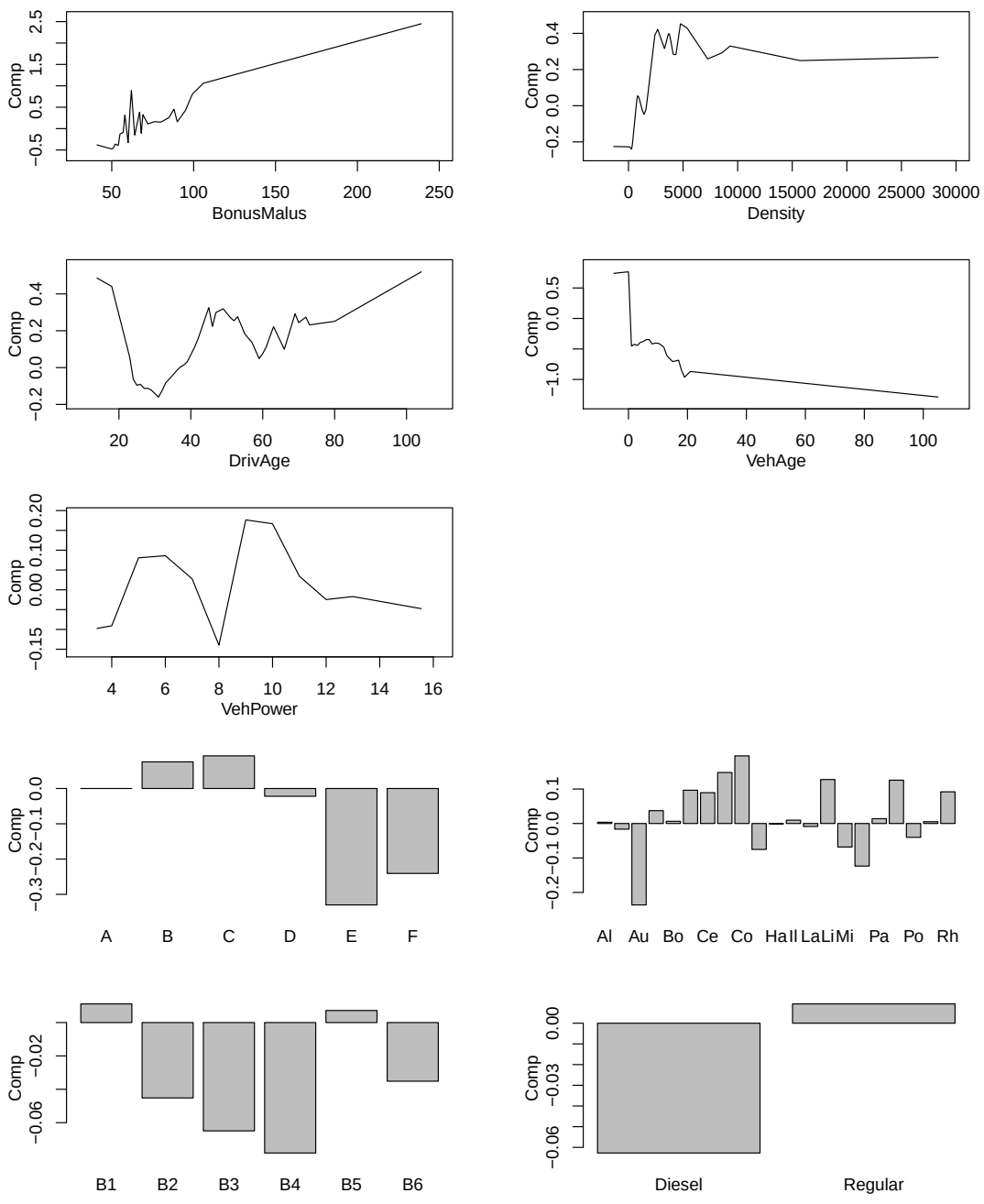

Figure 2: Component curves of AGLM

in the AGLM (for instance, in the case of the Poisson regression, we can say "it works multiplicatively".), while it is not easy to clearly explain how each curve in PDP contribute to the mean of the response variable.

Another interesting point to be discussed is whether these kinds of data-driven complicated component curves are acceptable in practices or not. However, it seems to rely on situations. Sometimes clear one-to-one relationships and high predictive accuracy might be enough, but sometimes not. For example, if there are some marketing or regulatory requirements on the feature, more smoothed curves or even constant components in some range might be desirable. However, in such cases, we can control component curves of the AGLM by using suitable binning or applying some post-processes like smoothing. The important thing is that it is clear how such modifications affect the response variable in the AGLM, as with the case of GLM.

6 Conclusions

In Chapter 4 and 5, We illustrated, both qualitatively and quantitatively, that our AGLM is a well-balanced model between interpretability and prediction accuracy as follows:

- Have a clear relationship between the features and the expected value of the response variable, as it is based on GLM. It exactly leads to high interpretability.
- It can be implemented to the data with a strong non-linear relationship between the features and the response variable, like GAM and the tree-type models. AGLM is comparable with these models in terms of prediction accuracy.

The second point above is achieved by automatic feature engineering with the O dummy variables and L variables. AGLM would be mathematically equivalent to Fussed Lasso with respect to quantitative variables. In addition, the L variables can be regarded as an extended model of Fused Lasso.

Furthermore, we developed an R package `aglm`, and we believe it provides excellent functionalities and flexibilities of modeling.

AGLM is developed with the idea of combining traditional GLM and the concept of recent techniques of data science so as to achieve both high interpretability and high predictive accuracy. We believe that these kinds of hybrid modeling methodologies will get more and more critical for actuarial practice in the future.

Appendix

We confirm the parity between the combination of the O dummy variables and the L1 regularization and the Fused Lasso indicated in Chapter 3.

For simplicity, consider a model with only one feature x with m possible values $\{1, ..., m\}$ and let $\{x_1, ..., x_n\}$ and $\{y_1, ..., y_n\}$ be observations and corresponding response variables respectively.

Then, use two types of dummy variables, $\{d_1(x_k), ..., d_m(x_k)\}$ and $\{d_1^O(x_k), ..., d_m^O(x_k)\}$, where $k = 1, ..., n$ and two parameters, $\{\gamma_1, ..., \gamma_m\}$ and $\{\beta_2, ..., \beta_m\}$, that satisfy $\gamma_i = \sum_{j=i+1}^{m} \beta_j$. We can obtain

$$\sum_{i=2}^{m} \gamma_i d_i(x_k) = \gamma_{x_k} = \sum_{j=x_k+1}^{m} \beta_j = \sum_{j=2}^{m} \beta_j d_j^O(x_k). \tag{12}$$

Consequently, take squared error function as an example, the following two optimization (minimization) problems are equivalent:

$$\min_{\gamma,\beta}\{-\frac{1}{2}\sum_{k=1}^{n}(y_k-\gamma_1-\sum_{i=2}^{m}\gamma_i d_i(x_k))^2+\lambda\sum_{i=2}^{m}|\gamma_{i-1}-\gamma_i|\}, \tag{13}$$

and

$$\min_{\gamma,\beta}\{-\frac{1}{2}\sum_{k=1}^{n}(y_k-\gamma_1-\sum_{j=2}^{m}\beta_j d_j^O(x_k))^2+\lambda\sum_{j=2}^{m}|\beta_j|\}. \tag{14}$$

The optimization problem (13) and (14) obviously represent the Fused Lasso and the Lasso, respectively. Therefore, the Fused Lasso can be run by the Lasso with the O dummy variables.

References

Alexandra Chouldechova, Trevor Hastie, and Vitalie Spinu. 2018. "Package 'Gamsel':Fit Regularization Path for Generalized Additive Models."

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Charpentier, Arthur. 2014. *Computational Actuarial Science with R*. CRC press.

Chouldechova, Alexandra, and Trevor Hastie. 2015. "Generalized Additive Model Selection." *arXiv Preprint arXiv:1506.03850*.

Devriendt, Sander, Katrien Antonio, Tom Reynkens, and Roel Verbelen. 2018. "Sparse Regression with Multi-Type Regularized Feature Modeling." *arXiv Preprint arXiv:1810.03136*.

Frees, Edward W, Richard A Derrig, and Glenn Meyers. 2014. *Predictive Modeling Applications in Actuarial Science*. Vol. 1. Cambridge University Press.

Friedman, Jerome H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics*, 1189–1232.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1.

Garavaglia, Susan, and Asha Sharma. 1998. "A Smart Guide to Dummy Variables: Four Applications and a Macro." In *Proceedings of the Northeast Sas Users Group Conference*, 43.

Gunning, David. 2017. "Explainable Artificial Intelligence (Xai)." *Defense Advanced Research Projects Agency (DARPA), Nd Web* 2.

Hastie, Trevor J. 2017. "Generalized Additive Models." In *Statistical Models in S*, 249–307. Routledge.

Hoerl, Arthur E, and Robert W Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12 (1): 55–67.

Jerome Friedman, Rob Tibshirani, Trevor Hastie, and Junyang Qian. 2019. "Package 'Glmnet': Lasso and Elastic-Net Regularized Generalized Linear Models."

J. Gertheiss, C. Oberhauser, S. Hogger, and G. Tutz. 2009. "Selection of Ordinally Scaled Independent Variables." *Applied Statistics Technical Report*, no. 62.

Nelder, John Ashworth, and Robert WM Wedderburn. 1972. "Generalized Linear Models." *Journal of the Royal Statistical Society: Series A (General)* 135 (3): 370–84.

Poon, Jacky HL, and others. 2019. "Penalising Unexplainability in Neural Networks for Predicting Payments Per Claim Incurred." *Risks* 7 (3): 1–11.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1): 267–88.

Tibshirani, Robert, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. "Sparsity and Smoothness via the Fused Lasso." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (1): 91–108.

Wüthrich, Mario V, and Michael Merz. 2019. "Yes, We Cann!" *ASTIN Bulletin: The Journal of the IAA* 49 (1): 1–3.

Yang, Yi, Wei Qian, and Hui Zou. 2018. "Insurance Premium Prediction via Gradient Tree-Boosted Tweedie Compound Poisson Models." *Journal of Business & Economic Statistics* 36 (3): 456–70.

Zou, Hui, and Trevor Hastie. 2005. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2): 301–20.