


Extracting information from textual descriptions for actuarial applications

Scott Manski, Kaixu Yang, Gee Y. Lee*  and Tapabrata Maiti

Michigan State University, East Lansing, USA

*Corresponding author. E-mail: leegee@msu.edu

(Received 28 April 2020; revised 13 September 2020; accepted 20 December 2020; first published online 02 March 2021)

Abstract

Initial insurance losses are often reported with a textual description of the claim. The claims manager must determine the adequate case reserve for each known claim. In this paper, we present a framework for predicting the amount of loss given a textual description of the claim using a large number of words found in the descriptions. Prior work has focused on classifying insurance claims based on keywords selected by a human expert, whereas in this paper the focus is on loss amount prediction with automatic word selection. In order to transform words into numeric vectors, we use word cosine similarities and word embedding matrices. When we consider all unique words found in the training dataset and impose a generalised additive model to the resulting explanatory variables, the resulting design matrix is high dimensional. For this reason, we use a group lasso penalty to reduce the number of coefficients in the model. The scalable, analytical framework proposed provides for a parsimonious and interpretable model. Finally, we discuss the implications of the analysis, including how the framework may be used by an insurance company and how the interpretation of the covariates can lead to significant policy change. The code can be found in the TAGAM R package (github.com/scottmanski/TAGAM).

Keywords: Actuarial modelling; Generalised additive models; GloVe; High dimensional; Lasso; Loss modelling; Risk analysis; Word embedding; Word similarity; Text analysis

1. Introduction

In actuarial practice, an important task of an insurance claims department is setting the case reserves for reported claims. The case reserve (case outstanding) for a given claim can be understood as the difference between the reported claim amount and the paid amount for an individual claim. The task is sometimes outsourced to third-party adjustors. For example, if the insurance company has made a partial payment of \$1,000 but expects to pay out an additional \$2,000 in the future, then \$2,000 is set as the case reserve. Note that the case reserve excludes incurred but unreported claims, for which a separate incurred but not reported (IBNR) reserve should be prepared. Some useful relationships are

$$\text{Reported Claims} = \text{Paid Claims} + \text{Case Reserves}$$

$$\text{Unpaid Claims} = \text{Case Reserves} + \text{IBNR}$$

$$\begin{aligned} \text{Ultimate Claims} &= \text{Reported Claims} + \text{IBNR} \\ &= \text{Paid Claims} + \text{Unpaid Claims} \end{aligned}$$

The concept of the case reserve is also explained in actuarial textbooks and manuals for ratemaking and reserving, such as Friedland (2010), and Werner & Modlin (2016). For the purpose of this

paper, the reader should understand that the case reserve is an approximation to the difference between the ultimate amount of the claim and the sum of the paid claims and the IBNR, given information available at the time of the report of the claim. Sometimes, the case reserve is set by the claims department of an insurance company, while in other cases the task is outsourced to an outside adjuster. Part of the information available to the claims department at the report time of the claim is a textual description of the claim. In this paper, we are interested in approaches that use the textual information regarding an insurance claim to predict the case reserve, by regressing the loss amount on a set of covariates derived from the textual description. Given a dataset of historic loss descriptions and ultimate loss amounts, an actuary may use the approach to improve the case reserving procedure. The problem is considered in Lee *et al.* (2019). This paper makes several extensions with sound statistical theory to make the actuarial work further automatic and reliable.

Part of the problem in this prediction task is that if we use a large number of keywords in forming the design matrix extracted from the textual descriptions of claims, the resulting problem is high dimensional in nature. In this case study, we use the framework of Lee *et al.* (2019) and analyse a dataset of loss descriptions and amounts, downloaded from the National Oceanic and Atmospheric Administration (NOAA).

For this analysis, a generalised linear model (GLM) may not be appropriate because the linearity assumption may not appropriately fit the data. To solve such a problem, we may consider using a non-parametric regression technique. A variety of non-parametric regression techniques have been developed, including but not limited to regression splines, kernel smoothing, neural networks, and generalised additive models (GAMs). Non-parametric regression has been applied in many areas, from modelling daily pollution in the UK (Wood *et al.*, 2017), to estimating relative risk for disease mapping of lung cancer (Dreassi *et al.*, 2014). See Simonoff (1996) for more details and examples of non-parametric regression. In this paper, we consider the GAM.

Hastie & Tibshirani (1986) proposed the generalised additive model that consists of the summation of smooth functions, allowing for the ability to capture the true, not necessarily linear, relationship. In the generalised additive model set-up, more information is needed to estimate each function as compared to the generalised linear model set-up. Therefore, the data must have many more observations than the number of covariates. In addition, when working with high-dimensional data, the scalability of the algorithm is also extremely important when considering a method. Our approach is motivated by these characteristics.

Considerable work has been done in efficiently estimating larger datasets using generalised additive models. Most recently, Wood *et al.* (2017) developed a method for estimating GAMs with the number of coefficients of order 10^4 , and observations up to 10^8 . This method reduces the number of matrix operations, utilises parallelisation, and reduces the memory necessary by marginal discretisation of the model covariates. Li & Wood (2019) extended this work by proposing an alternative method of calculating $\mathbf{X}'\mathbf{W}\mathbf{X}$ where \mathbf{X} is a model matrix and \mathbf{W} a diagonal or tri-diagonal matrix, which results in a 30-fold reduction in computational time. Previous works include Marra & Wood (2011) and Wood *et al.* (2015). Code for these methods are found in the R package `mgcv`, Wood (2019).

While the aforementioned GAM results provide for a scalable algorithm, a hindrance of GAM is the restriction on the number of covariates. Considering the GLM, there are several methods for combating the high-dimensionality issue, with the most notable one being lasso by Tibshirani (1996). Similar to the GLM, the lasso maximises the likelihood, but instead has an additional L_1 penalty term. This term is typically referred to as the shrinkage term. Extensions of the lasso have also been developed, including but certainly not limited to, group lasso from Yuan & Lin (2006) and adaptive lasso from Zou (2006). The group lasso is applied to variables with group-like structure, and it uses a slightly altered penalty term where each variable in a group is penalised equally. This is particularly important due to the group-like structure induced by the basis expansion used in the estimation of the generalised additive model. The adaptive lasso simply applies a weight to

each coefficient in the penalty term, with these weights typically estimated through ordinary least squares or lasso. Wang & Leng (2008) combined these extensions to formulate the adaptive group lasso and showed the ability of the method to identify the true model consistently.

Our proposed method is a three-step approach consisting of the following steps: (1) weight calculation by group lasso; this first step uses a preliminary model to generate the weights used in the second step. (2) The shrinkage step; this step uses adaptive group lasso, using the weights generated from the first step, in order to have a consistent model selection. (3) The smoothing step; this step uses the reduced problem from step 2 in order to estimate the smooth functions corresponding to the remaining covariates. The approach combines the adaptive group lasso dimension reduction technique with the scalable GAM algorithm.

In summary, this paper proposes a method for analyzing high-dimensional data with a non-linear relationship between the predictors and the response. The method is easy to apply and is a general approach that can be applied in various contexts. To demonstrate its use, we apply it to the textual data analysis problem in an actuarial context. We propose using word embedding matrices and cosine similarities to convert textual data into numeric data, which can be used as the design matrix for the proposed method. The rest of the paper proceeds in the following order: in section 2, the dataset used for the analysis is summarised. In section 3.1, the details of the model is explained. Section 3.2 explains how the model parameters can be estimated using a three-step approach, and section 3.3 summarises the approach in the form of an algorithm. Section 4 presents the results from the data analysis. Section 5 presents some implications and discussions regarding our model, and section 6 concludes the paper with closing remarks. The Appendix presents the theoretical foundation for our method.

2. Data and Pre-processing

For our analysis, we utilise the publicly available NOAA Storm Events Database. The analysis is performed on property loss amounts at the event level, using storm event observations involving textual descriptions of the events. The data are collected over time; however, we use a cross-sectional model in this paper in order to focus on the relationship between the textual information and the response. Only *Thunderstorm Wind* events taking place in Michigan and from 2000 to 2018 are considered for the analysis. We have selected a specific sample for demonstration, but in general we have found that the result is similar regardless of the sample chosen, as long as there is a reasonable number of observations found in the sample. In general, our method would work well when there is a non-linear relationship between the predictor and the covariates, and the problem is high dimensional.

For losses spanning a long period, inflation should be taken into consideration in order for the model to be used in practice. This can be accomplished in many different ways. One approach is to use trending methods as in traditional actuarial science practices. Alternatively, one may use statistical models that take the effect of inflation into consideration. For simplicity of demonstration, in this paper, we assume the effect of inflation is not our primary concern. The resulting prediction can be adjusted for inflation using trending methods as needed.

For validations, the dataset is divided into training and validation datasets. The reason we use this dataset is because it contains relatively clean, lengthy descriptions of losses from storm events in the United States each year, along with the property and crop damage amount estimates. These damage amounts are initial estimates of the losses and hence are different from the ultimate loss amounts. Yet, the structure of the data is identical to that available to a claims adjuster and hence is a good test dataset for the analytical framework explained in this paper. Another advantage of this dataset is that it is publicly available, allowing dissemination and reproducibility to be easy.

Each event is recorded with an event narrative. An example of an observation with an estimated property damage of \$10,000 has an event narrative that reads: *Roof damage was incurred to a barn*

Table 1 Summary statistics for the log(loss) for the training and validation datasets

	<i>N</i>	Min	Mean	SD	Max
Training	2,353	2.30	8.97	1.44	17.03
Validation	126	6.21	8.78	1.56	14.00

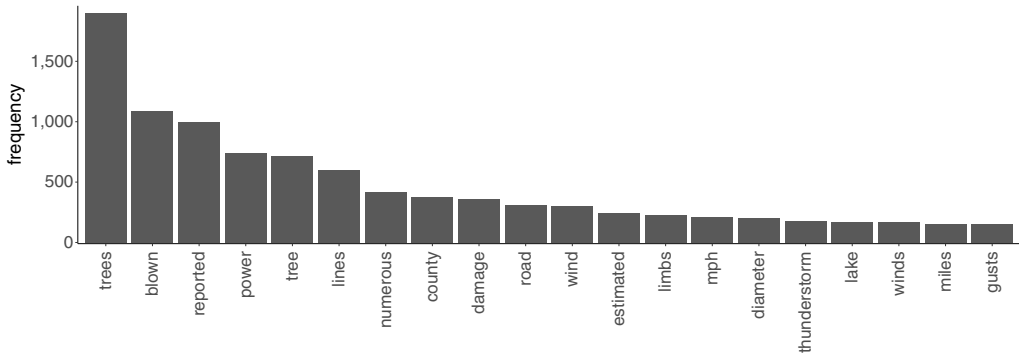


Figure 1. Frequency for the most common words.

six miles northwest of Mason due to a severe thunderstorm wind gust and a large tree limb was blown down in South Lansing.

Figure 1 shows the most common words in the descriptions of the losses. Stop-words such as *a*, *the*, and *and*, etc., have been removed. Notice that the word *trees* is most frequent in the descriptions. A few of the most common words are typically used to describe what is happening to *trees*, such as *blown* and *wind*. In addition, several of the other most common words like *power*, *lines*, *damage*, and *outages* are used to describe the results of downed trees.

There are a total of 2,353 observations in the training set, with 126 observations in the validation set. As previously mentioned, the claim descriptions are quite lengthy, with an average of 16.8 words per description. There are a total of 2,642 unique words used in the dataset. To capture only relevant words, stop words, numbers, and words that only occurred once were removed, resulting in 1,998 words. Table 1 provides summary statistics for the log(loss) for the training and validation datasets.

In order to better understand the relationship between the words in the claim description and the property loss amount, each word is represented by a vector. Recent advancements in word embedding models have made it possible to obtain these representations easily. We utilise the 300-dimensional word embeddings developed by the authors of Pennington *et al.* (2014). To form the design matrix, we follow the framework described by Lee *et al.* (2019). That is, for two words with vector representations **a** and **b**, respectively, the cosine similarity is defined as:

$$\text{sim}_{\text{cos}}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2} \tag{1}$$

Moreover, for a given phrase, let $D = (\mathbf{b}_1, \dots, \mathbf{b}_S)$ where each $\mathbf{b}_i, i \in \{1, \dots, S\}$ is a word in the phrase. Then define the cosine similarity between a word **a** and a phrase *D* as:

$$\text{sim}_{\text{cos}}(\mathbf{a}, D) = \max_{s=1, \dots, S} (\text{sim}_{\text{cos}}(\mathbf{a}, \mathbf{b}_s)) \tag{2}$$

In this way, we construct a matrix of cosine similarities $\mathbf{X}_{n \times p_n}$ where *n* is the number of observations and p_n is the number of unique words used. Let *W* be the vector of unique words with

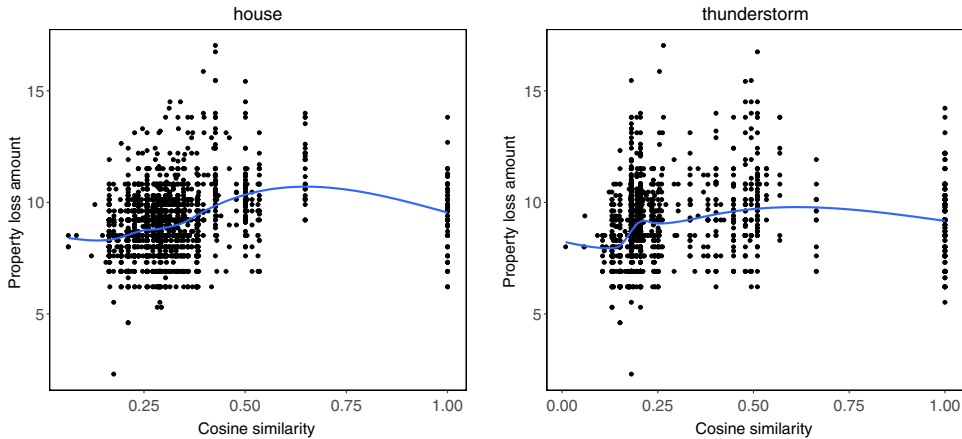


Figure 2. Cosine similarity against property loss for *house* and *thunderstorm*.

length p_n , and let \mathbf{D} be the list of descriptions with length n , where each element in the list is a vector D_i containing the words used in description i . Then,

$$X_{ij} = \text{sim}_{\text{cos}}(W_j, D_i) \quad \text{for } i \in \{1, \dots, n\}, j \in \{1, \dots, p_n\} \quad (3)$$

Each value in the matrix is now continuous and restricted to $[-1, 1]$. Figure 2 shows the relationship between cosine similarity and property loss for *house* and *thunderstorm*. From the figure, we see that the relationship between cosine similarity and property loss is non-linear in nature and therefore a generalised additive model is appropriate.

Regarding the text data processing method, cosine similarities is just one method that works. Our contribution in this paper is mainly the development of a predictive model after the pre-processing is done. The reader may observe that there exists noise in the predictors, and in Lee *et al.* (2019), this noise was dealt with a cut-off value for the cosine similarities. In this paper, our focus is on the high dimensionality of the problem, and the cut-off values are set to $\epsilon = 0$ by default. Our approach is not meant to be the best candidate in terms of the pre-processing step. Also, we believe that the predictability is not influenced much by the cut-off, or the ϵ value in Lee *et al.* (2019) In Figure 2, the observations with cosines of 1 are not really outliers, but in fact observations with high similarity with the word. Hence, these are very important observations, perhaps more so than the noise corresponding to low cosine similarities. One may imagine there are some data missing between the cosines of 1 and those with small cosines. Our method tries its best under this restriction.

Note that the predictive power of each variable is not known in advance, so a variable selection technique is appropriate first to identify the variables which are statistically meaningful. We would like to emphasise that first, GAM has been proven to work in the high-dimensional set-up, with solid theoretical foundations; see Huang *et al.* (2010), Yang & Maiti (2020). It is guaranteed that the GAM consistently estimates the parameters for the model. Second, the GAM provides more interpretability than “black-box” machine learning algorithms such as random forest, or neural networks. We are able to plot each estimated selected function and gain insights from the results.

3. Methodology

In this section, we describe our methodology in specific terms. Section 3.1 specifies the model for the high-dimensional generalised additive model. Section 3.2 describes the three-step approach to the parameter estimation. Section 3.3 summarises the approach in the form of an algorithm.

3.1 High-dimensional generalised additive model

We consider the generalised additive model:

$$\mu_i = E[y_i | \mathbf{X}_i] = g^{-1} \left(\sum_{j=1}^{p_n} f_j(X_{ij}) \right) \tag{4}$$

where the link function corresponds to that of the corresponding exponential family distribution. For each of the n independent observations, the density function is given by:

$$f_{y_i} = c(y) \exp \left[\frac{y\theta_i - b(\theta_i)}{\phi} \right], \quad 1 \leq i \leq n, \quad \theta_i \in \mathbb{R} \tag{5}$$

We assume that a matrix of explanatory variables is given. Let's call it $\mathbf{X}_{n \times p_n}$, and use the notation $\mathbf{X} = (\mathbf{X}_1^T, \mathbf{X}_2^T, \dots, \mathbf{X}_n^T)^T$. We have

$$\mathbf{X}_{n \times p_n} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p_n} \\ X_{21} & X_{22} & \dots & X_{2p_n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np_n} \end{bmatrix} \tag{6}$$

Thus, n is the number of observations and p_n is the number of explanatory variables available. We assume that the parameter $0 < \phi < \infty$ is known. Without loss of generality, let $\phi = 1$. Also, we assume that the density of y_i depends on \mathbf{X}_i via the structure:

$$\theta_i = \sum_{j=1}^{p_n} f_j(X_{ij}) \tag{7}$$

where θ_i are defined in equation (5). Assume that the additive components belong to the Sobolev space $W_2^d([a, b])$. According to Schumaker (1981), see pp. 268–270, there exists B-spline approximation:

$$f_{nj}(x) = \sum_{k=1}^{m_n} \beta_{jk} \phi_k(x), \quad 1 \leq j \leq p \tag{8}$$

with $m_n = K_n + l$, where K_n is the number of internal knots and $l \geq d$ is the degree of the splines. Generally, it is recommended that $d = 2$ and $l = 4$, that is, cubic splines:

$$\max_{1 \leq k \leq K+1} |\xi_k - \xi_{k-1}| = O(n^{-\nu}) \tag{9}$$

For a practical overview of the B-spline basis function, the reader may refer to Wood (2017), section 5.3.3, starting from p. 204. We want to write

$$f_{nj}(X_{ij}) = \sum_{k=1}^{m_n} \Phi_{ik}^{[j]} \beta_{jk} \tag{10}$$

for some value $\Phi_{ik}^{[j]}$. We call Φ our design matrix and denote the elements of the design matrix ϕ_{it} , for $i = 1, \dots, n$ and $t = 1, \dots, q_n$. We also denote

$$\Phi_{ij} = \left(\Phi_{i1}^{[j]}, \Phi_{i2}^{[j]}, \dots, \Phi_{im_n}^{[j]} \right)^T, \quad \text{for } i = 1, \dots, n, \text{ and } j = 1, \dots, p_n \tag{11}$$

Under this framework, the response variable is related to the covariate X_{ij} via

$$f_j(X_{ij}) = \Phi_{ij}^T \beta_j, \quad i = 1, \dots, n, \quad j = 1, \dots, q \tag{12}$$

where β_j is the coefficient corresponding to the j -th explanatory variable. We may see that β_j must be a length m_n vector, since the j -th spline contains m_n parameters. Our methodology is related to Chouldechova & Hastie (2015), yet we use a three-step procedure. We are looking for the parameters for:

$$g \{E[y_i|X_i]\} = \beta_0 + \sum_{j=1}^{p_n} f_{nj}(X_{ij}) = \beta_0 + \sum_{j=1}^{p_n} \Phi_{ij}^T \beta_j = \beta_0 + \Phi_i \beta \tag{13}$$

where we have used the notation Φ_i to denote the i -th row of Φ , and $\beta = (\beta_1^T, \beta_2^T, \dots, \beta_{p_n}^T)^T$, where some of the β_j 's are zero, while others are non-zero. The approach in Chouldechova & Hastie (2015) is to minimise the penalised log-likelihood:

$$-\frac{1}{n} \ell(\beta) + \lambda_{n2} \sum_{j=1}^{p_n} \sqrt{\beta_j^T S_j \beta_j} + \frac{1}{2\phi} \sum_{j=1}^{p_n} \lambda_{n3j} \beta^T D_j \beta \tag{14}$$

where $\ell(\beta)$ is the log-likelihood for an exponential family distribution:

$$\begin{aligned} \ell(\beta) &= \sum_{i=1}^n \left[y_i \left(\sum_{j=1}^{p_n} \sum_{k=1}^{m_n} \Phi_{ik}^{[j]} \beta_{jk} \right) - b \left(\sum_{j=1}^{p_n} \sum_{k=1}^{m_n} \Phi_{ik}^{[j]} \beta_{jk} \right) \right] \\ &= \sum_{i=1}^n \left[y_i \left(\Phi_i^T \beta \right) - b \left(\Phi_i^T \beta \right) \right] \end{aligned} \tag{15}$$

The hope is that the second term in equation (14) induces zeros into groups of coefficients, while the last term imposes smoothness into the “surviving” coefficients. Here, S_j is an identity matrix of dimension m_n and D_j is a constraint matrix to impose smoothness into the estimated functions f_j . There are several practical difficulties with this approach:

- When p_n is large, or in other words when the problem dimension is large, there are too many λ_{n3j} tuning parameters to estimate. Wood (2017) discusses algorithms for large n cases but does not talk about cases where p_n is large.
- Theory behind selecting the tuning parameters λ_{n3j} discussed in Wood (2017) is no longer directly applicable because of the extra group lasso-type penalty term.
- Implementing the coordinate descent algorithm, which brings in sparsity into β , becomes tricky with the smoothing penalty. Usually, fast algorithms for lasso-type estimators with GLMs are implemented by locally approximating the likelihood with a Taylor’s approximation at each iterative step, yet the extra penalty term makes this tricky.
- Estimating the coefficients may take a very long time, especially when the number of explanatory variables p_n is large, as in the application we consider in this paper.

Hence, in order to keep the estimation procedure scalable for large p_n (and hence large q_n), we propose a three-step approach to the estimation problem for the model (13). The first step of the approach is to perform a group lasso estimation with the first and second terms of equation (14). The second step uses the resulting coefficient estimates to perform an adaptive group lasso estimation of the parameters. The third and final step uses the non-zero coefficients obtained from the second step to induce smoothness into the implied spline function $f_{nj}(\cdot)$, for each non-zero function f_{nj} . These steps are formalised in the following section.

Moreover, to provide a statistical validation, we present both the numerical results in section 4 and the theory for the estimated functions in the Appendix A, which works as another support of our proposed three-stage approach. We aim at validating two things: the variables selected are consistent and the estimators are consistent with respect to the unknown true functions.

3.2 Learning framework: the three-stage approach

In this section, we explain how the parameters for the model presented in the previous section can be estimated using a three-stage approach. The first is a group lasso step, where the weights for the second step are determined. The second step is an adaptive group lasso step, where the weights obtained from the first step are used to reduce the problem dimension. The reason why we need to separate the first and second step is because the second step ensures selection consistency. The third step is the smoothing step, where a smoothness penalty is used to obtain the correct parameters for the additive model. The input to the three-step method is a matrix of cosine similarities, and the output is a set of smooth functions corresponding to the explanatory variables that have been selected from the procedure.

3.2.1 Stage 1 – group lasso

Define the objective function to be

$$L(\boldsymbol{\beta}; \lambda_{n1}) = -\frac{1}{n} \sum_{i=1}^n \left[y_i \left(\boldsymbol{\Phi}_i^T \boldsymbol{\beta} \right) - b \left(\boldsymbol{\Phi}_i^T \boldsymbol{\beta} \right) \right] + \lambda_{n1} \sum_{i=1}^{p_n} \|\boldsymbol{\beta}_j\|_2 \tag{16}$$

Let $\hat{\boldsymbol{\beta}}$ be the optimiser for (16), or in other words,

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{p_n \cdot m_n}} L(\boldsymbol{\beta}; \lambda_{n1}) \tag{17}$$

3.2.2 Stage 2 – adaptive group lasso

Define the objective function to be

$$L_a(\boldsymbol{\beta}; \lambda_{n2}) = -\frac{1}{n} \sum_{i=1}^n \left[y_i \left(\boldsymbol{\Phi}_i^T \boldsymbol{\beta} \right) - b \left(\boldsymbol{\Phi}_i^T \boldsymbol{\beta} \right) \right] + \lambda_{n2} \sum_{j=1}^{p_n} w_{nj} \|\boldsymbol{\beta}_j\|_2 \tag{18}$$

where the weights depend on the screening stage group lasso estimator:

$$w_{nj} = \begin{cases} \|\hat{\boldsymbol{\beta}}_j\|_2^{-1} & \text{if } \|\hat{\boldsymbol{\beta}}_j\|_2 > 0 \\ \infty & \text{if } \|\hat{\boldsymbol{\beta}}_j\|_2 = 0 \end{cases} \tag{19}$$

Numerically, the weights are set to a large number, for the case when $\|\hat{\boldsymbol{\beta}}_j\|_2 = 0$.

Let $\hat{\boldsymbol{\beta}}_{AGL}$ be the optimiser for (18). In other words,

$$\hat{\boldsymbol{\beta}}_{AGL} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{p_n \cdot m_n}} L_a(\boldsymbol{\beta}; \lambda_{n2}) \tag{20}$$

Let \hat{S}_n be the subset of $\{1, \dots, p\}$, such that the j th coefficient of $\boldsymbol{\beta}_{AGL}$ with $j \in \hat{S}_n$ are non-zero. Thus, the second-stage estimates are sparse, meaning that the coefficients are zero for some j . This reduces the coefficient size in the third stage.

3.2.3 Stage 3 – the smoothness penalty

Let $\hat{\boldsymbol{\Phi}}^{\hat{S}_n}$ be the matrix consisting of columns from $\boldsymbol{\Phi}$ corresponding to the set \hat{S}_n . Let $\boldsymbol{\beta}_{\hat{S}_n}$ be in $\mathbb{R}^{\hat{S}_n \cdot m_n}$, where $\hat{S}_n = |\hat{S}_n|$. Define the objective function to be

$$L_{sm}(\boldsymbol{\beta}; \lambda_{n3}) = -\frac{1}{n} \sum_{i=1}^n \left[y_i \left(\boldsymbol{\beta}^T \boldsymbol{\Phi}_i^{\hat{S}_n} \right) - b \left(\boldsymbol{\beta}^T \boldsymbol{\Phi}_i^{\hat{S}_n} \right) \right] + \frac{1}{2\phi} \sum_{j \in \hat{S}_n} \lambda_{n3j} \boldsymbol{\beta}_j^T \mathbf{D}_j \boldsymbol{\beta}_j \tag{21}$$

where $\lambda_{n3} = (\lambda_{n31}, \lambda_{n32}, \dots, \lambda_{n3p_n})$. Let $\hat{\beta}_{sm}$ be the optimiser for (21). In other words,

$$\hat{\beta}_{sm} = \arg \min_{\beta \in \mathbb{R}^{m \times n}} L_{sm}(\beta; \lambda_{n3}) \quad (22)$$

Since the problem of dimension has been reduced, the third-step estimation may be performed using existing generalised additive models routines, using $\hat{\beta}_{AGL}$ as the initial guess for the penalized iteratively reweighted least squares (P-IRLS) procedure. The tuning parameters λ_{n3} may be obtained by generalised cross-validation or restricted maximum likelihood (REML) as described in Wood (2017).

In variable selection, the smoothness penalty term is actually not required. The intuition behind this is that a function has to have enough signal strength to be considered significant, while the wiggly estimations are close to the true functions in terms of signal strength, though they might be more wiggly around the smooth functions. Therefore, the first two steps are able to provide a reasonable set of variables as the final predictors. However, estimation without smoothness penalty can lead to overfitting, thus the third step is there to remedy this issue. As the results in Huang *et al.* (2010) and Yang & Maiti (2020) show, the first two steps consistently identify the significant variables with probability tending to 1, thus the third stage can be considered to perform a low-dimensional GAM on a reasonable set of predictors.

3.2.4 Tuning parameters

Each stage has a tuning parameter, λ_{n1} , λ_{n2} , and λ_{n3} , respectively. The selection of λ_{n1} and λ_{n2} can greatly influence the performance of the model and the efficiency of the algorithm. Larger values of λ_{n1} and λ_{n2} will lead to an over-simplified model with faster computation time, while smaller values will lead to an over-fitted model with slower computation time. To find the “sweet spot,” cross-validation is used to determine λ_{n1} and λ_{n2} . The tuning parameters λ_{n3} is obtained by generalised cross-validation or REML as described in Wood (2017).

3.3 Learning algorithm and its implementation

We now discuss the implementation of the method using R. For stage 1 and stage 2, we utilise functions from the `gglasso` package (Yang & Zou, 2017) and for stage 3 we utilise functions from the `mgcv` package (Wood, 2019). The code is provided in an R package at github.com/scottmanski/TAGAM.

3.3.1 Stage 1 – Group lasso

The `gglasso` function is modified such that we loop through the grid of λ_{n1} values, but once the number of non-zero coefficients is greater than n , the algorithm is stopped. By doing so, we ensure that we will be able to execute stage 3.

3.3.2 Stage 2 – Adaptive group lasso

The implementation of stage 2 is very similar to that of stage 1, except for the addition of the weights. In order to incorporate the weights, let $\beta'_j = w_{nj}\beta_j$ for each $j \in \{1, \dots, p_n\}$. Then equation (18) can be written as:

$$L_a(\beta'; \lambda_{n2}) = -\frac{1}{n} \sum_{i=1}^n \left[y_i \left(\sum_{j=1}^{p_n} \frac{1}{w_{nj}} \Phi_i^{[j]T} \beta'_j \right) - b \left(\sum_{j=1}^{p_n} \frac{1}{w_{nj}} \Phi_i^{[j]T} \beta'_j \right) \right] + \lambda_{n2} \sum_{j=1}^{p_n} \|\beta'_j\|_2 \quad (23)$$

Table 2 Summary statistics for the final model. The residual degree of freedom (DF) comes from the estimated degrees of freedom from the GAM, and the mean squared prediction error (MSPE) is the out-of-sample mean squared prediction error.

K_n	l	λ_{n1}	λ_{n2}	\hat{S}_n	Residual DF	Deviance	MSPE
4	2	0.0005255074	0.0001063902	149	2167.387	70.7%	1.016

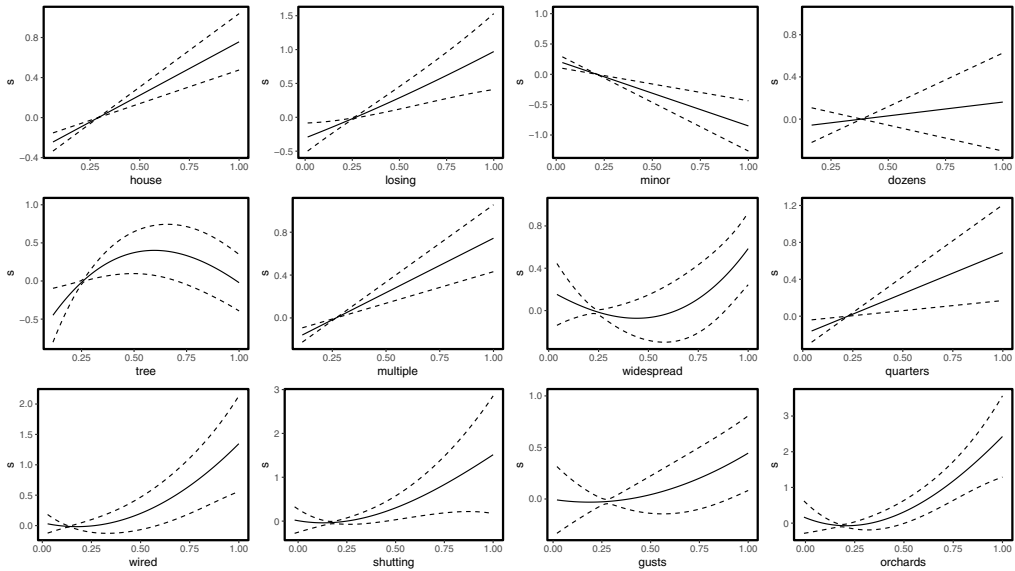


Figure 3. Function estimates for several covariates.

3.3.3 Stage 3 – The smoothness penalty

The *mgcv* package is used to implement stage 3. In the *mgcv* package, there is a *gam* function and a *bam* function, with the former designed for smaller datasets and the latter designed for much larger datasets. In this analysis, we utilise *bam*. To increase the computational efficiency, we also choose to have the function discretise the data following the method described in Wood *et al.* (2017).

4. Data Analysis

In this section, we discuss the results of our model. Table 2 provides information for the final model. As previously mentioned, 1,998 words appeared in the dataset and were considered as possible covariates. For the model, we chose to use the penalised regression spline. Stage 1 effectively reduced the number of covariates to 261, and stage 2 further reduced the number of words to 149. While the number of functions to interpret may seem cumbersome, the final model is relatively simple compared to the number of possible covariates that could have been in the model.

Figure 3 shows the estimated functions for several covariates. All of the function estimates have a few characteristics in common. For smaller cosine similarity values, the estimated functions are approximately zero. We expect this because smaller cosine similarities between a word and a phrase indicates that the word has very little meaning in common with the phrase. For large cosine similarity values, the 95% credible interval for the functions becomes wider as compared to cosine similarity values around 0.2. This is also expected simply due to the lack of observations for higher cosine similarities.

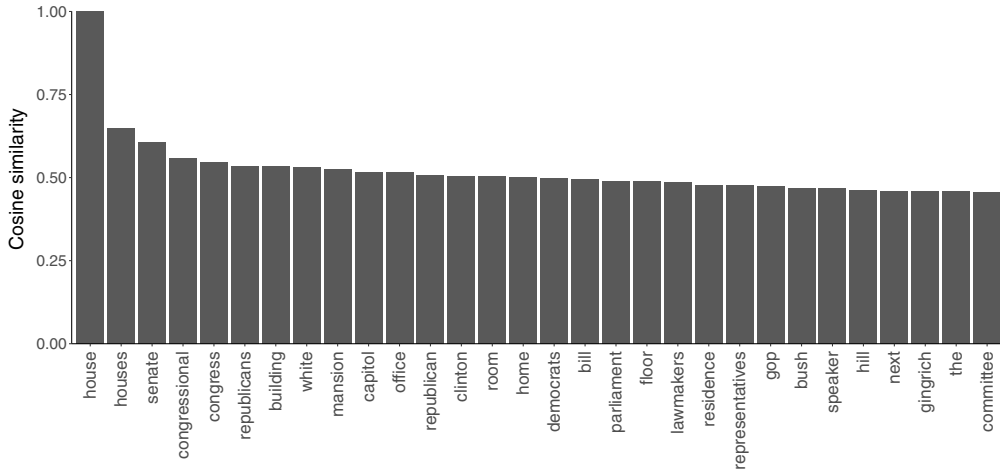


Figure 4. Words with the highest cosine similarity with *house*.

The function estimates help us understand the relationship between a word, its related words, and the property loss amount. Many of these estimated functions seem to follow our intuition. For example, *house*, *losing*, *widespread*, *gusts*, and *tree* are all words that would typically be associated with property loss. Words with the highest cosine similarity to *house* are shown in Figure 4. Most of these related words are types of homes. The cosine similarities capture the likelihood of a word being close to a particular concept, and the relationship is not meant to be perfect. One may imagine the results showing up in a search engine. Typing in a keyword allows for related documents to be searched from the internet; however, sometimes irrelevant contents may appear in the search result as well. This problem is acknowledged in Lee *et al.* (2019), and the problem is partially coped by setting a cut-off value for the cosine similarities in Lee *et al.* (2019). From the function estimate, we see that an incident involving a house results in higher property loss than that of an incident involving offices or apartments, in general.

While many function estimates obviously follow our intuition, there are some that seem harder to interpret. Words like *quarters*, *shutting*, and *orchards* all seem unrelated to property loss. To shed some light on this issue, we look at a sentence from a description that includes *quarters*; *two eyewitnesses in Covington reported hail greater than the size of quarters during the peak of the storm*. The use of *quarters* here is related to the size of hail. It is expected that larger hail will lead to larger property loss. Words related to *quarters* include *nickel* and *dime*, which are also used to describe hail size. In a similar way, we find out that *shutting* is referring to the closure of major roadways. In the case of *orchards*, several observations involved damage to apple orchards. With Michigan producing the third most apples of any state, it is clear why damage to apple orchards results in large property loss.

The model also performed well with out-of-sample prediction. Figure 5 shows the predicted property loss amounts against the true loss amounts for the validation sample. The Spearman correlation for the validation set is 76.06%, while the Spearman correlation for the training dataset is 80.30%.

To measure the stability of the method, for a selected year, the model was trained using the previous years and tested on data from the selected year. This was completed for each year from 2001 to 2018. This resulted in an average mean squared prediction error of 1.34 with a standard error of 0.123. Using a lasso model increases each of these values by about 8%, respectively. The three-stage method selected a more parsimonious model as compared to the single-step lasso model, resulting in greater model stability.

Table 3. Comparison of models.

Model	Spearman correlation (%)	MSPE	Gini index
Three-stage model	76.06	0.996	0.076
Random forest	73.64	1.016	0.064
Lasso	73.38	1.074	0.058
Indicator model	72.59	1.180	0.049

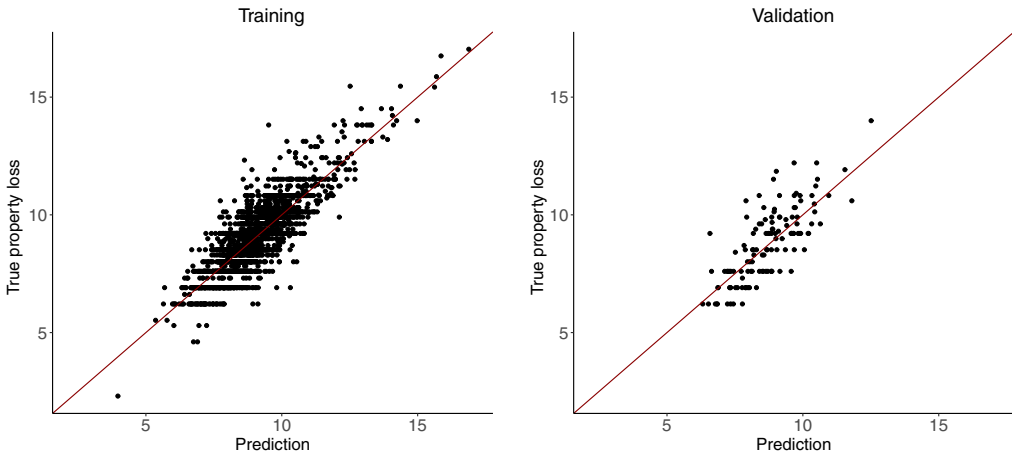


Figure 5. Predicted property loss amounts against the true property loss amounts for the training and validation samples. The Spearman correlations are 80.61% and 76.06%, respectively.

Several additional models were fit to the data, and the out-of-sample results are compared. The candidate models are random forest, lasso, and the indicator model. First, a random forest model has been fit to the cosine similarities. Second, a Lasso model has been fit to the cosine similarities, with no basis expansion. This model can be seen as a baseline model, fixing the cosine similarities. Third, the lasso model has been fit to indicator variables of whether each word is present in the description. Each word in the dataset has its own indicator variable, and if the word appears in the event description, then that variable is 1. According to the results shown in Table 3, we see that the three-stage model performs best in terms of Spearman correlation, MSPE, and Gini index.

5. Discussion

We have presented an analytical method for analyzing losses due to storm events in relation to their textual descriptions. The fact that losses may be predicted more accurately with textual information implies that the case reserving procedure may be improved significantly. The traditional approach to case reserving is to take the average amount of the reported losses, yet this does not take advantage of the heterogeneity of information contained within the initial report of a loss to an insurance company. The new method allows for a more accurate prediction of the ultimate loss to be indemnified for a specific reported loss.

Being able to explain the factors that contribute to higher or lower severity of losses by selecting the relevant keywords from a set of words allows the actuarial analyst to avoid manually selecting the keywords needed for the textual risk analysis. This technique may be useful, especially when the number of words describing the loss is large, or statistically the problem is high

dimensional. The analyst may also be able to understand the factors that relate to high losses using the selected covariates, and this may help mitigate future losses.

In addition, these factors that contribute to higher severity of property loss can indicate areas needing improvement in the way they protect against various weather events. For example, events involving *orchards* resulted in high property loss, illustrating the need for additional preventative measures to protect the apple trees during a thunderstorm.

The fact that a simple three-step approach allows for the regression selection problem to be solved easily using existing routines in the R programming language.

The two-step approach is proven to have selection consistency in the high-dimensional set-up, for example, see Huang *et al.* (2010). In section 3 of Huang *et al.* (2010), the screening consistency and estimation convergence rate of the first-step estimator are established, but no selection consistency is guaranteed. Similar results are in Yang & Maiti (2020). The second step improves the selection result of the first step with a better convergence rate and is proven to have selection consistency. The predictors selected by the two-step approach is more reliable and stable. (see Lemma 1 in the Appendix.) Thus using the law of total probability and the fact that probabilities are less than or equal to one, we are able to show that the difference between performing the third-step estimation on the true variables and on the selected variables tends to 0 as $n \rightarrow \infty$.

In Theorem 1, the estimation consistency of the third step is shown. An important property of predictive modelling, the prediction error, is a direct result of the estimation error. In our model, we have the expected prediction error:

$$E_X \|\hat{y} - y\|_2^2 \leq \sum_{j=1}^s \|\hat{f}_j - f_j\|_2^2 + \epsilon^2 + \epsilon_{embed}^2$$

where three components are here: the estimation error, the random error, and the word embedding error. Since random error is not under control and embedding error depends on the word embedding algorithm, bounding the estimation error is equivalent to bounding the prediction error, under mild conditions on the design matrix.

Similarly, in the confidence interval of the third step, the difference between conditioning on correct selection and not conditioning on correct selection is bounded by a negligible term, which is the probability of not selecting the correct variables and disappears as $n \rightarrow \infty$. Although, the theory of confidence band has not been established in this high-dimensional set-up, following a referee's comment, a small simulation study has been performed to support this argument. The simulation verifies the 95% confidence intervals for the function estimates. Samples of size 400 were used with 50 covariates each with a randomly selected function. The breakdown of true functions is: Exponential (12), Linear (7), Logarithmic (5), Polynomial (5), Sinusoidal (8), and Zero (13).

After fitting the three-stage model, for the confidence interval of each estimated function, we calculate the empirical coverage rate, that is, we determine the proportion of the time that the confidence interval contains the true function. To do this, we choose a point x_0 and determine if the confidence interval for the estimated function contains the true function at point x_0 . This is repeated for 1,000 choices of x_0 . We average this value across all estimated functions to find empirical coverage rate for the model. This process was repeated for 100 iterations and the average proportion (with standard error) is 0.9612 (0.00172). These results empirically verify the validity of point-wise confidence intervals obtained from the three-step approach.

6. Concluding Remarks

In this paper, we consider a general high-dimensional text analysis problem and propose a three-stage approach by adopting modern statistical methods. Stage 1 and 2 effectively reduced the high-dimensional problem to one that mgcv can handle. The use of stage 1 and 2 to reduce the

problem instead of utilising a subject matter expert allows for simple replicability of the process. We showed how the use of cosine similarities from textual descriptions can provide interpretable results when predicting property loss. While there are many other possible applications in risk analysis, our framework could also be applied in the classification of users on a social networking site based on their posts, prediction of a company's change in stock price from related articles, and caller scam classification based on call transcripts.

The approach may also be applied in general to problems where non-linear effects of a large number of continuous explanatory variables must be understood in relation to the response. We have focused on the log-normal case of the response, yet the method is general enough to be applied to non-normal responses, including responses following a gamma distribution or Poisson distribution. Future work may focus on these specific cases.

Acknowledgements. We thank the editor and two anonymous reviewers, who reviewed our manuscript and provided constructive comments to improve this paper.

References

- Chouldechova, A. & Hastie, T.** (2015). Generalized additive model selection. arXiv preprint, arXiv:1506.03850v2 [stat.ML].
- Dreassi, E., Ranalli, M.G. & Salvati, N.** (2014). Semiparametric m-quantile regression for count data. *Statistical Methods in Medical Research*, **23**, 591–610. Available online at the address <https://doi.org/10.1177/0962280214536636>. PMID: 24847899.
- Friedland, J.** (2010). *Estimating Unpaid Claims Using Basic Techniques*. Casualty Actuarial Society, Arlington County, VA, USA.
- Hastie, T. & Tibshirani, R.** (1986). Generalized additive models. *Statistical Science*, **1**, 297–310.
- Huang, J., Horowitz, L. & Wei, F.** (2010). Variable selection in nonparametric additive models. *Annals of Statistics*, **38**, 2282–2313.
- Lee, G.Y., Manski, S. & Maiti, T.** (2019). Actuarial applications of word embedding models. *ASTIN Bulletin*, **50**(1), 1–24.
- Li, Z. & Wood, S.N.** (2019). Faster model matrix crossproducts for large generalized linear models with discretized covariates. *Statistics and Computing*. Available online at the address <https://doi.org/10.1007/s11222-019-09864-2>.
- Marra, G. & Wood, S.** (2011). Practical variable selection for generalized additive models. *Computational Statistics and Data Analysis*, **55**, 2372–2387.
- Pennington, J., Socher, R. & Manning, C.D.** (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Available online at the address <http://www.aclweb.org/anthology/D14-1162>.
- Schumaker, L.** (1981). *Spline Functions: Basic Theory*. John Wiley & Sons, New York.
- Simonoff, J.S.** (1996) *Smoothing Methods in Statistics*. Springer, New York, NY.
- Tibshirani, R.** (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, **58**(1), 267–288.
- Wang, H. & Leng, C.** (2008). A note on adaptive group lasso. *Computational Statistics & Data Analysis*, **52**, 5277–5286.
- Werner, G. & Modlin, C.** (2016). *Basic Ratemaking*. Casualty Actuarial Society, Arlington County, VA, USA.
- Wood, S.** (2019) *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. Available online at the address <https://CRAN.R-project.org/package=mgcv>. R package version 1.8-31.
- Wood, S.N.** (2017) *Generalized Additive Models: An Introduction with R*, 2nd edition. CRC Press, Taylor & Francis Group, Boca Raton, FL, USA.
- Wood, S.N., Goude, Y. & Shaw, S.** (2015). Generalized additive models for large data sets. *Journal of the Royal Statistical Society, Series C*, **64**, 139–155.
- Wood, S.N., Li, Z., Shaddick, G. & Augustin, N.H.** (2017). Generalized additive models for gigadata: Modeling the U.K. black smoke network daily data. *Journal of the American Statistical Association*, **112**, 1199–1210.
- Yang, K. & Maiti, T.** (2020). Ultra high dimensional generalized additive model: unified theory and methods. Available online at the address <https://arxiv.org/abs/2008.06773>.
- Yang, Y. & Zou, H.** (2017). *gglasso: Group Lasso Penalized Learning Using a Unified BMD Algorithm*. Available online at the address <https://CRAN.R-project.org/package=gglasso>. R package version 1.4.
- Yuan, M. & Lin, Y.** (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B*, **68**, 49–67.
- Zou, H.** (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418–1429.

A. Appendix: Theory of the Third-Stage Estimator

In this section, we will provide statistical foundation for the proposed approach. For this reason, we derive the convergence rate for our third-stage estimator. This will establish statistical consistency of our procedure. In Yang & Maiti (2020), the following result for the second-stage estimator has been established.

Lemma A.1 (Yang & Maiti, 2018) *The adaptive group lasso consistently selects the true active predictors in probability, that is, the estimator $\hat{\beta}_{AGL}$ satisfies*

$$\mathbb{P} \left(\|\hat{f}_{AGLj}(x)\|_2 > 0, j \in T \text{ and } \|\hat{f}_{AGLj}(x)\|_2 = 0, j \in T^c \right) \rightarrow 1 \tag{A.1}$$

The results states that with proper choices of λ_{n1} and λ_{n2} , the adaptive group lasso consistently selects the true non-zero predictors. This theorem guarantees the selection consistency of the three-stage algorithm, since the variable selection is done in the second stage and the third stage does not do variable selection. It is important for an algorithm to select the correct subset of variables for the model built on them to work.

With similar assumptions, assume we have

Assumption 1. *The true functions f_1, \dots, f_{s_n} has smoothness order o_n , that is,*

$$\int_a^b f_j''(x)^2 dx \asymp o_n$$

where $a_n \asymp b_n$ means there exist constants c and d such that

$$c \leq \frac{a_n}{b_n} \leq d$$

Then, we have

Theorem 1. *Under assumptions 1 and assumptions in Yang & Maiti (2020), for tuning parameters $\lambda_{n31}, \dots, \lambda_{n3s_n}$, we have*

$$\|\hat{f}_{sm} - f_{sm}^0\|_2^2 = O_p \left(s_n \gamma_2^{-2s_n} m_n \frac{\log(s_n m_n)}{n} \right) + O_p(s_n^2 \gamma_2^{-2s_n} m_n^{-2d}) + O_p \left(\sum_{j \in \hat{S}_n} \lambda_{n3j} o_n \right) \tag{A.2}$$

where γ_0 and γ_2 are assumed bounds parameters in eigenvalues of X , see Yang & Maiti (2020).

Theorem 1 shows the rate of convergence of the third-stage estimator. There are three terms in the convergence rate: the estimation error, the spline approximation error, and the regularisation error. The greater the o_n , the less the λ_{n3} is, thus the product will not change. This theorem guarantees that with proper choice of parameters, the estimated functions are consistent estimators of the true functions that describe the relationship between the variables and the response.

Proof. Consider the third step, where we have the smoothness penalty. Define the event:

$$\mathcal{S}_n = \{\hat{S}_n = S\}$$

The previous lemma showed that

$$\mathbb{P}(\mathcal{S}_n) \rightarrow 1 \text{ as } n \rightarrow \infty$$

From now on, let us condition on the event \mathcal{S}_n . For convenience, we suppress the notations $\hat{\beta}_{sm}$, β_{sm}^0 , and $\Phi^{\hat{S}_n}$ and denote them with $\hat{\beta}$, β^0 , and Φ .

To study the characteristics of the smoothness term S_j , where

$$\int_a^b f_j''(x)^2 dx = \int_a^b \phi''(x)\phi''(x)^T dx = \beta_j^T S_j \beta$$

without loss of generality, consider the case that the knots are evenly distributed on the interval $[a, b]$, since changing the length of the intervals does not change the shape of the B-splines but the span and height (Schumaker, 1981). In the following calculations, we normalise the interval $[a, b]$ to $[0, Kl_n]$, where each interval has length l_n . According to Huang *et al.* (2010), assume the constant length of the interval satisfies $l_n = O(n^{-\nu})$ with $0 < \nu < 0.5$. The k th cubic B-spline basis can be derived from definition:

$$B_{k,4}(x) = \begin{cases} \frac{x^3}{6l_n^3} - \frac{kx^2}{2l_n^2} + \frac{k^2x}{2l_n} - \frac{k^3}{6}, & l_n k \leq x \leq l_n(k+1), \\ \frac{-3x^3}{6l_n^3} + \frac{(9k+10)x^2}{6l_n^2} - \frac{7k^2+16k+6}{6l_n} + \frac{k^3+2k^2-2k-2}{6}, & l_n(k+1) \leq x \leq l_n(k+2), \\ \frac{3x^3}{6l_n^3} - \frac{(9k+20)x^2}{6l_n^2} + \frac{9k^2+42k+34}{6l_n} - \frac{k^3+8k^2+14k+10}{6}, & l_n(k+2) \leq x \leq l_n(k+3), \\ \frac{-x^3}{6l_n^3} + \frac{(k+2)x^2}{6l_n^2} - \frac{3k^2+20k+32}{6l_n} + \frac{k^3+10k^2+32k+32}{6}, & l_n(k+3) \leq x \leq l_n(k+4), \\ 0, & \text{o.w} \end{cases}$$

and we have $\phi(x) = \{B_{k,4}(x), k = 1, \dots, m_n\}$. Taking derivative, we have the second derivative of the basis function satisfies

$$B''_{k,4}(x) = O(l_n^{-2}) = O(n^{2\nu})$$

Therefore, the elements:

$$s_{j,ik} = O(n^{3\nu}) \text{ for } j = 1, \dots, p \text{ and } i, k = 1, \dots, m_n \text{ where } s_{j,ik} \in S_j$$

and equals exactly zero if $|i - k| > 3$. As a direct result, the eigenvalue of the matrix S_j is bounded from above by $O(n^{3\nu})$ and from below by some constant. Similarly, if we use a quadratic B-spline, the elements $s_{j,ik}$ are bounded from above by $O(n^{4\nu})$ and from below by some constant.

Then, we begin the convergence rate part. For a converging sequence N_n such that $\|\hat{\beta} - \beta^0\|_2 \leq N_n$, define $t = N_n / (N_n + \|\hat{\beta} - \beta^0\|_2)$, then consider the convex combination $\beta^* = t\hat{\beta} + (1 - t)\beta^0$. We have $\beta^* - \beta^0 = t(\hat{\beta} - \beta^0)$, which implies

$$\|\beta^* - \beta^0\|_2 = t\|\hat{\beta} - \beta^0\|_2 = \frac{N_n\|\hat{\beta} - \beta^0\|_2}{N_n + \|\hat{\beta} - \beta^0\|_2} \leq N_n \tag{A.3}$$

This means β^* is within a small distance from β^0 and we are safe to use Taylor expansion. Moreover, if we have

$$\|\beta^* - \beta^0\|_2 \leq R_n$$

then

$$\frac{N_n}{N_n + \|\hat{\beta} - \beta^0\|_2} \|\hat{\beta} - \beta^0\|_2 \leq R_n$$

Choosing N_n to be greater than R_n , we have

$$\|\hat{\beta} - \beta^0\|_2 \leq 2R_n$$

Therefore, it is sufficient to derive the convergence rate for β^* .

Consider the Taylor expansion:

$$\begin{aligned} & -\frac{1}{n} \sum_{i=1}^n \left[y_i (\beta^{*T} \Phi_i) - b(\beta^{*T} \Phi_i) \right] \\ = & -\frac{1}{n} \sum_{i=1}^n \left[y_i (\beta^{0T} \Phi_i) - b(\beta^{0T} \Phi_i) \right] - \left(\frac{1}{n} \sum_{i=1}^n \left[y_i \Phi_i - b'(\beta^{0T} \Phi_i) \Phi_i \right] \right)^T (\beta^* - \beta^0) \\ & + \frac{1}{2n} \sum_{i=1}^n (\beta^* - \beta^0)^T \Phi_i^T b''(\beta^{**} \Phi_i) \Phi_i (\beta^* - \beta^0) \\ =: & -\frac{1}{n} \sum_{i=1}^n \left[y_i (\beta^{0T} \Phi_i) - b(\beta^{0T} \Phi_i) \right] - \frac{1}{n} (y - \mu^0)^T \Phi (\beta^* - \beta^0) \\ & + \frac{1}{2n} (\beta^* - \beta^0)^T \Phi^T \Sigma(\beta^{**}) \Phi (\beta^* - \beta^0) \end{aligned}$$

where μ^0 is the expectation of y at β^0 and $\Sigma(\beta^{**})$ is the covariance matrix of y evaluated as β^{**} which is located on the line segment joining β^0 and β^* .

By the definition of β^* and convexity, we have

$$\begin{aligned} & -\frac{1}{n} \sum_{i=1}^n \left[y_i (\beta^{*T} \Phi_i) - b(\beta^{*T} \Phi_i) \right] + \sum_{j \in \hat{\Delta}_n} \lambda_{n3j} \beta_j^{*T} D_j \beta_j^* \\ \leq & -\frac{1}{n} \sum_{i=1}^n \left[y_i (\beta^{0T} \Phi_i) - b(\beta^{0T} \Phi_i) \right] + \sum_{j \in \hat{\Delta}_n} \lambda_{n3j} \beta_j^{0T} D_j \beta_j^0 \end{aligned}$$

Combine this with the Taylor expansion result, we have

$$\begin{aligned} & \frac{1}{2n} (\beta^* - \beta^0)^T \Phi^T \Sigma(\beta^{**}) \Phi (\beta^* - \beta^0) \\ \leq & \frac{1}{n} (y - \mu^0)^T \Phi (\beta^* - \beta^0) + \sum_{j \in \hat{\Delta}_n} \lambda_{n3j} \left[\beta_j^{0T} D_j \beta_j^0 - \beta_j^{*T} D_j \beta_j^* \right] \\ \leq & \frac{1}{n} |(y - \mu)^T \Phi (\beta^* - \beta^0)| + \frac{1}{n} |(\mu^0 - \mu)^T \Phi (\beta^* - \beta^0)| + \sum_{j \in \hat{\Delta}_n} \lambda_{n3j} \left[\beta_j^{0T} D_j \beta_j^0 - \beta_j^{*T} D_j \beta_j^* \right] \\ \leq & \frac{1}{n} |(y - \mu)^T \Phi (\beta^* - \beta^0)| + \frac{1}{4n} (\beta^* - \beta^0)^T \Phi^T \Sigma(\beta^{**}) \Phi (\beta^* - \beta^0) + O\left(s_n^2 m_n^{-2d}\right) \\ & + \sum_{j \in \hat{\Delta}_n} \lambda_{n3j} \left[\beta_j^{0T} D_j \beta_j^0 - \beta_j^{*T} D_j \beta_j^* \right] \end{aligned}$$

where the second inequality comes from norm inequality, the third inequality comes from Cauchy–Swarchz inequality, and μ is the expectation of y given f^0 . Rearranging the inequality, we have

$$\begin{aligned} & \frac{1}{4n} (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0)^T \Phi^T \Sigma (\boldsymbol{\beta}^{**}) \Phi (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0) \\ & \leq \frac{1}{n} | (y - \mu)^T \Phi (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0) | + O\left(s_n^2 m_n^{-2d}\right) + \sum_{j \in \hat{S}_n} \lambda_{n3j} \boldsymbol{\beta}_j^{0T} D_j \boldsymbol{\beta}_j^0 \\ & \leq \frac{1}{8n} (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0)^T \Phi^T \Sigma (\boldsymbol{\beta}^{**}) \Phi (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0) + \frac{2}{n} \|\Sigma^{1/2} (\boldsymbol{\beta}^{**}) (y - \mu)\|_2^2 \\ & \quad + \sum_{j \in \hat{S}_n} \lambda_{n3j} \boldsymbol{\beta}_j^{0T} D_j \boldsymbol{\beta}_j^0 + O\left(s_n^2 m_n^{-2d}\right) \end{aligned}$$

where the inequality is by Cauchy–Swarchz inequality. Consider the penalty matrix D_j who has entries $d_{j,ik} = 1$ if $i = k = 1$, $d_{j,ik} = 2$ if $i = k \neq 1$ and $d_{j,ik} = -1$ if $|i - k| = 1$. The matrix is a constant matrix, thus each $\boldsymbol{\beta}_j^{0T} D_j \boldsymbol{\beta}_j^0$ is of the order $O(o_n)$. Rearranging the terms and by Concentration inequality, see for example Yang & Maiti (2020), we have

$$\frac{1}{8n} (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0)^T \Phi^T \Sigma (\boldsymbol{\beta}^{**}) \Phi (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0) = O\left(s_n m_n \frac{\log(s_n m_n)}{n}\right) + O\left(\sum_{j \in \hat{S}_n} \lambda_{n3j} o_n\right) + O\left(s_n^2 m_n^{-2d}\right)$$

By Remark 2.1 in Yang & Maiti (2020), we have

$$\frac{\gamma_0 c_1 \gamma_2^{2s_n}}{8m_n} \|\boldsymbol{\beta}^* - \boldsymbol{\beta}^0\|_2^2 \leq \frac{1}{8n} (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0)^T \Phi^T \Sigma (\boldsymbol{\beta}^{**}) \Phi (\boldsymbol{\beta}^* - \boldsymbol{\beta}^0)$$

Combine the above result with the condition event S_n , we have conditioning on S_n :

$$\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^0\|_2^2 = O_p\left(s_n \gamma_2^{-2s_n} m_n \frac{\log(s_n m_n)}{n}\right) + O\left(s_n^2 \gamma_2^{-2s_n} m_n^{-2d}\right) + O\left(\sum_{j \in \hat{S}_n} \lambda_{n3j} o_n\right)$$

By the inequality that

$$P(A) = P(A|B)P(B) + P(A|B^C)P(B^C) \leq P(A|B) + P(B^C)$$

consider $S_n = B^C$, we have

$$\|\boldsymbol{\beta}^* - \boldsymbol{\beta}^0\|_2^2 = O_p\left(s_n \gamma_2^{-2s_n} m_n \frac{\log(s_n m_n)}{n}\right) + O_p\left(s_n^2 \gamma_2^{-2s_n} m_n^{-2d}\right) + O_p\left(\sum_{j \in \hat{S}_n} \lambda_{n3j} o_n\right)$$

Combine this with the argument at the beginning of the proof, we have

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^0\|_2^2 = O_p\left(s_n \gamma_2^{-2s_n} m_n \frac{\log(s_n m_n)}{n}\right) + O_p\left(s_n^2 \gamma_2^{-2s_n} m_n^{-2d}\right) + O_p\left(\sum_{j \in \hat{S}_n} \lambda_{n3j} o_n\right)$$

Cite this article: Manski S, Yang K, Lee GY and Maiti T (2021). Extracting information from textual descriptions for actuarial applications, *Annals of Actuarial Science*, 15, 605–622. <https://doi.org/10.1017/S1748499521000026>