Nonlife Insurance Risk Classification Using Categorical Embedding

Peng Shi, PhD, ACAS, FSA
University of Wisconsin-Madison
Email: pshi@bus.wisc.edu

Kun Shi, PhD, FCAS
Southwest Airlines Co.
Email: kun.shi@gmail.com

Abstract

This article presents several actuarial applications of categorical embedding in the context of nonlife insurance risk classification. In nonlife insurance, many rating factors are naturally categorical and often the categorical variables have a large number of levels. The high cardinality of categorical rating variables presents challenges in the implementation of traditional actuarial methods. Categorical embedding that is proposed in the machine learning literature for handling categorical variables has recently received attention in actuarial studies. The method is inspired by the neural network language models for learning text data and maps a categorical variable into a real-valued representation in the Euclidean space. Using a property insurance claims data set, we demonstrate the use of categorical embedding in three applications. The first shows how embeddings are used to construct rating classes and calculate rating relativities for a single insurance risk. The second concerns predictive modeling for multivariate insurance risks and emphasizes the effects of dependence on tail risks. The third focuses on pricing new products where transfer learning is used to gather knowledge from existing products.

Keywords: actuarial applications, categorial embedding, nonlife insurance, risk classification, transfer learning

1 Introduction

Risk classification is at the core of underwriting and ratemaking, the two basic functions in nonlife insurance. Underwriting and ratemaking are closely associated with each other: the former deals with the selection of risks and the latter concerns pricing the accepted risks. In risk classification, an insurer determines the rating factors that discriminate between policyholders and assigns policyholders into homogeneous categories. Policyholders within a class have similar risk profiles and are charged the same price. A refined risk classification system leads to cream skimming in underwriting and actuarial fair premium in ratemaking.

In this study, we examine an unique problem in nonlife insurance risk classification. Specifically, many rating factors in a risk classification system are naturally categorical, and often the categorical rating variables have a large number of levels with most common examples being postal code in homeowner insurance and vehicle model and make in automobile insurance. The high cardinality in the categorical rating variables imposes challenges in the implementation of the traditional actuarial methods, in particular, the generalized linear models (GLMs). Motivated by these observations, we introduce the method of categorical embedding to insurance risk classification and show its value to various actuarial practices.

Over decades of practice, GLMs have become a standard method in actuarial toolkits for pricing insurance products (see Haberman and Renshaw (1996) and De Jong et al. (2008)). Actuarial community is comfortable with GLMs because of its strong connection with the traditional actuarial pricing technique known as minimum bias methods (see Brown (1988) and Mildenhall (1999)). However, GLMs have some difficulties when categorical rating factors have a large number of levels. First, the high cardinality of rating factors leads to a high-dimensional design matrix, which requires an unrealistic amount of computational resource. Second, for a given data, there is a higher likelihood of insufficient data in some categories of the rating variable, which attributes to higher uncertainty in parameter estimates and prediction. Third, the relationship between different levels of the rating variable is usually ignored. In the case there are subgroups among the large number of levels of the rating variable, traditional methods cannot automatically reflect the similarities among them. To address these issues, different studies have proposed using credibility or in general information-sharing method in the GLM setting. For instance, Ohlsson and Johansson (2006) treated multi-level rating factors as a random quantity and used the classic Bühlmann-Straub framework to drive the credibility estimation; Klinker (2011) employed generalized linear mixed-effects models to obtain the shrinkage estimator and discussed its application in ratemaking; Frees and Lee (2015) considered regularization method in the case of insufficient data and used regularized regression for rating endorsement in property insurance.

Setting apart from the existing studies, we employ the method of categorical embedding to learn the effects of categorical rating variables of high cardinality on insurance risk outcomes. Thereafter, we use term "risk" to refer any uncertain outcome of a policyholder. For example, it could represent the aggregate losses for a policyholder over the contract year, or it could represent the losses of a policyholder from a single coverage or a single peril. Categorical embedding is a deep learning

method that maps a categorical variable into a real-valued representation in the Euclidean space. The method can be formulated as a deep neural network with an extra embedding layer between the input and hidden layers and thus the mapping can be automatically learned in the supervised training process. The seminal idea of categorical embedding is due to the neural network language models for learning text data (Guo and Berkhahn (2016)). In recent years, deep learning models using artificial neural networks have been developed for automated text processing in that the models can directly handle unstructured data and perform feature engineering as part of learning process (see, for example, Kim (2014), Lai et al. (2015), Vaswani et al. (2017), and Devlin et al. (2019)). Categorical variables share similarity with text data in that words can be interpreted as a variable with a large number of categories with each word in a dictionary corresponding to a category. The difference is that one data point of a categorical variable has only one level, while a sentence of words has multiple levels with an informative order.

Since categorical embedding can be automatically learned by a deep neural network, it is sensible to view the method as an artificial neural network with a special deep learning architecture to handle categorical inputs. This perspective is appealing when prediction is the primary interest of the study. In fact, neural networks have been extensively used for function approximation because of its impressive learning ability to predict complex nonlinear relationships between input and output variables (Goodfellow et al. (2016)). In case of high cardinality, categorical embedding reduces the number of parameters substantially, which mitigates potential overfitting and therefore is expected to improve prediction.

Recently, the method of categorical embedding has been introduced to actuarial applications. For instance, Perla et al. (2021) applied the embedding techniques for categorical variables of geographical regions in mortality forecasting. Kuo and Richman (2021) discussed embedding and attention methods in predictive modeling and compared results using flood insurance data. Vincent et al. (2022) considered using adversarial learning to promote fairness in pricing nonlife insurance contract. The primary contribution of our paper is to identify and present three novel actuarial applications of categorical embedding in the context of nonlife insurance risk classification. In the first application, we look into the standard single risk setting and show how embeddings are used to create rating classes and compute associated relativity. The second application examines the context of multivariate insurance risks, where we formulate the joint distribution of dependent risks and emphasize the effect of dependence on tail risks. The third one is regarding pricing new products with sparse data. We employ transfer learning to obtain knowledge on rating variables from existing products.

Despite the appealing predictive aspect, we stress that the interest of categorical embedding is often the embedding itself rather than the prediction of the outcome. An embedding is essentially a dense representation in the form of numeric vectors in the low-dimensional embedding space. For text data, each word is represented by a numeric vector. For instance, Lee et al. (2020) demonstrated using word embedding in claims triage - a key component in insurance claim management. For categorical data, each category is represented by a numeric vector. In this work, we will

demonstrate using categorical embedding in insurance risk classification. For this line of studies, the embedding in lieu of the predicted outcome is the output of primary interest from the trained neural network. In categorical embedding models, the embedding layer is the unique feature in the deep learning architecture that differentiates from the usual neural network architecture and warrants the automatic computing of the embeddings for categorical variables.

The rest of the paper is structured as follows: Section 2 introduces the insurance claims dataset and describes the key features of the rating factors and the risk outcome. Section 3 gives a brief introduction to neural networks that is required for understanding the categorical embedding by deep learning methods. Section 4 provides a detailed description of the categorical embedding method using deep neural networks. Section 5 identifies three actuarial applications where we demonstrate the value of categorical embedding with an property insurance claim data set. Section 6 concludes the article.

2 Data

The insurance claims dataset is obtained from the local government property insurance fund of Wisconsin (hereinafter referred to as the fund). Functioning as a commercial-line insurance provider, the fund provides property insurance coverage for local government entities, e.g. municipal buildings, schools, and libraries. We examine the building and contents insurance that covers damage to both physical structures and items inside including equipments, furniture, inventory, supplies and fixtures. The data is an extended version of the one analyzed by Shi and Yang (2018) and Yang and Shi (2019).

There are 1,110 entities with each observed during years 2006-2013. We consider risk outcome at the policy-year level where one thinks of each local government entity as a policyholder. Claims data are aggregated to each policy year by perils for individual policyholders, despite the fact that each policy could provide coverage for multiple buildings at different locations. The final data contains observations at both policy and peril level in a total of 8,880 policy years.

Table 1 provides a description of rating variables, among which, coverage and deductible amounts are numeric, and entity type and county code are categorical. Categorical embedding is performed for discrete rating variables, and the primary interest is the county code which has a large number of categories.

Table 1: Description of rating variables

Variable	Description
EntityType	Entity type of the policyholder (city, county, school, town, or village, or other)
County	County code of the geographical region of the property (72 categories)
Coverage	Amount of insurance coverage
Deductible	Amount of deductible

In this study, we consider a binary outcome variable that measures the claim frequency of

policyholders. Specifically, the outcome equals one indicating the entity has at least one claims during the policy year and zero otherwise. In addition, we observe the cause of loss for each claim from which we create the binary claim frequency outcome for each peril. Table 2 shows the overall empirical probability of claims as well as the claim probability by peril. There is a 29% chance that a random selected policyholder will have claims over the year regardless of peril type, and the chances are 16%, 12%, and 12% for fire, water, and other perils respectively. We also report in the table the empirical claim probability by entity type. There is significant variation of claim probability across entity type, suggesting that the entity type provides a sensible basis for risk classification. To visualize the relation between claim frequency and coverage, we exhibit in Figure 1 the box plot of the amount of coverage by the binary claim frequency. Similar relations are observed for the overall and the peril-wise claim frequency. As anticipated, a larger coverage amount suggests higher exposure to risk and thus a higher likelihood of claims.

Table 2: Claim probability by peril and by entity type

	Overall	Entity Type					
		City	County	School	Town	Village	Misc
Claim Probability	0.291	0.522	0.743	0.302	0.077	0.263	0.116
Peril Type							
Fire	0.158	0.333	0.457	0.133	0.044	0.138	0.057
Water	0.124	0.225	0.464	0.114	0.020	0.103	0.039
Other	0.116	0.206	0.362	0.145	0.023	0.068	0.037
Number of Obs	8,880	1,247	541	2,469	1,510	2,117	996

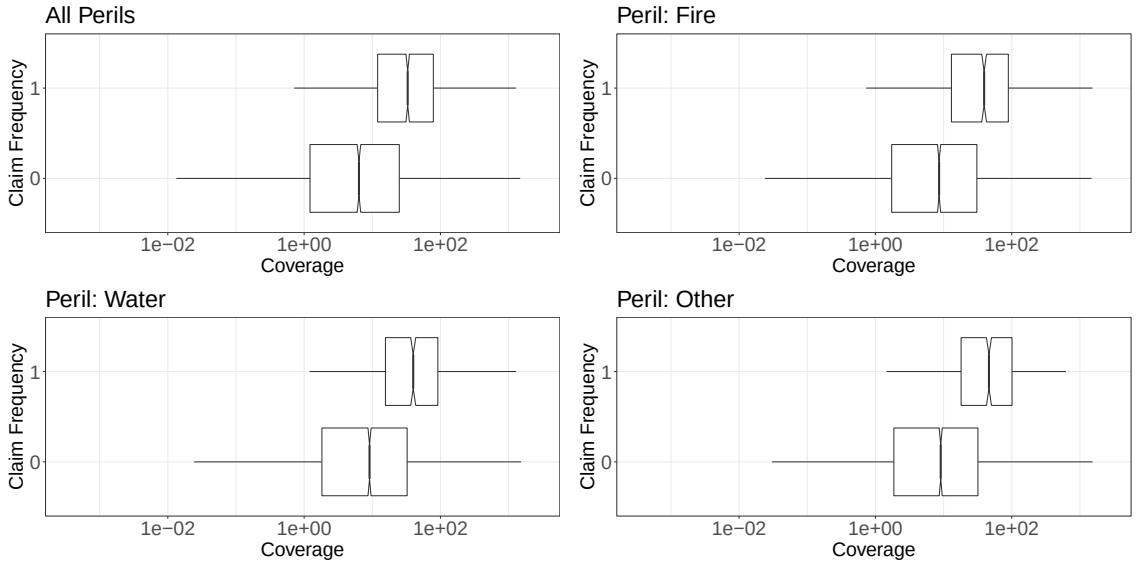

Figure 1: Relation between the amount of coverage and claim frequency by peril type.

Finally, we emphasize that the binary claim outcomes are not independent with each other.

To provide some evidence, we present in Table 3 the two-way contingency table for the claim frequency for each pair of perils. Consider the pair of fire and water perils. On one hand, among policyholders without fire claims, 9% (675/7474) have at least one water claims. In contrast, among the policyholders with at least one fire claims, 30% (426/1406) have water claims. On the other hand, among policyholders without water claims, 13% (980/7779) have at least one fire claims. In contrast, among the policyholders with at least one water claims, 39% (426/1101) have fire claims. The analysis suggests positive relationship between the two claim outcomes and similar patterns are observed for all other pairs of perils. The positive association among peril-wise claim frequency outcomes is further confirmed by the large χ^2 and G statistics reported in the table. One explanation of the positive correlation is the common factor of thunderstorm. In our data, fire claims are due to damages caused by fire including lightning, and water claims could be both weather and non-weather related. Policyholders with higher exposure to thunderstorms are likely to observe claims due to either lightning or water. Another explanation is the risk control, i.e. a policyholder who exercises more risk control could reduce both fire and water hazards.

Table 3: Contingency table for peril-wise claim frequency

			Water			
		0	1	Total		
Fire	0	6,799	675	7,474	χ^2-statistic	490.84
	1	980	426	1,406	G-statistic	398.37
	Total	7,779	1,101	8,880		
			Other			
		0	1	Total		
Fire	0	6,850	624	7,474	χ^2-statistic	478.41
	1	1,002	404	1,406	G-statistic	385.51
	Total	7,852	1,028	8,880		
			Other			
		0	1	Total		
Water	0	7,067	712	7,779	χ^2-statistic	358.15
	1	785	316	1,101	G-statistic	283.55
	Total	7,852	1,028	8,880		

3 A Brief Review of Neural Networks

Neural networks are engineered computational models inspired by mammals' brain - a biological nervous system. The mammalian brain contains between 100 million and 100 billion biological neurons, depending on the species. Because of the structure and functional properties of these interconnected neurons, the brain is able to perform complex and computationally demanding tasks such as face recognition and body movement. The artificial neural network was developed to emulate the learning ability of the biological neuron system.

The initial concept of artificial neural networks traces back to the 1940s (see McCulloch and

Pitts (1943)). To date the neural network has become a powerful machine learning tool and been successfully employed in a wide range of applications (see Schmidhuber (2015) and LeCun et al. (2015)). The basic mathematical model of neural networks consists of a series of layers, including the input, hidden, and output layers. For illustration, Figure 2 provides a graphical representation of the feedforward neural network. Each layer contains a set of artificial neurons also known as perceptron and the number of neurons could vary by layers. The left panel shows a single-layer perceptron where inputs are directly fed to the output layer, and the right panel shows a 2-layer perceptron where inputs pass through a hidden layer before fed to the output layer.

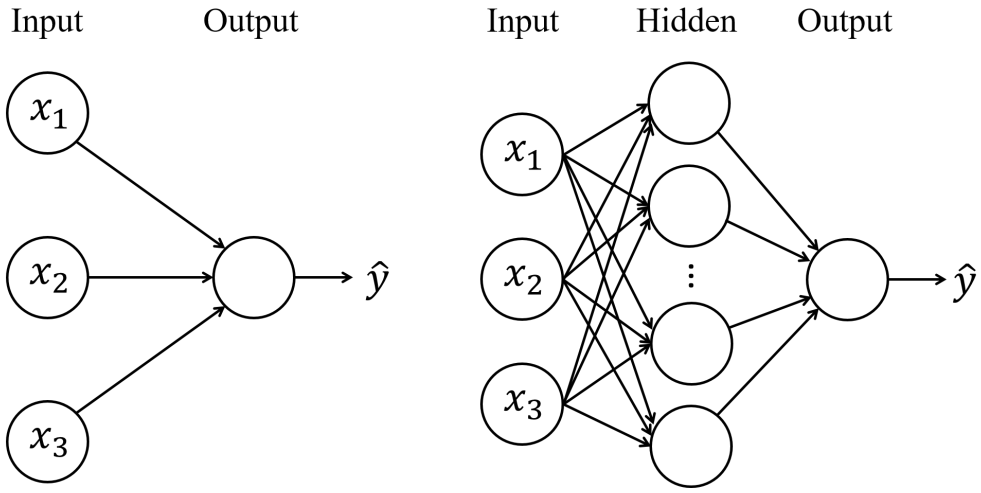

Figure 2: Graphical illustration of artificial neural networks. The left panel shows a single-layer perceptron, and the right panel shows a 2-layer perceptron.

A deep neural network can be constructed by adding more hidden layers between the input layer and the output layer. The number of hidden layers is referred to as the depth of the deep neural network. Deep learning methods are essentially the neural networks with many hidden layers. We refer readers to Goodfellow et al. (2016) for a comprehensive discussion of deep learning and neural networks. Despite that each individual neuron has little learning capability, when functioning cohesively together, neural networks have shown tremendous learning ability and computational power to predict complex nonlinear relationships. Because of this, neural networks have been extensively used for function approximation, and they can be applied to either regression (continuous output) or classification (discrete output) problems. Let y be the output variable and $\boldsymbol{x} = (x_1, \ldots, x_p)'$ be the p-dimensional set of input variables. In a regression problem, a neural network approximates the unknown function $f(\boldsymbol{x})$ that relates the output and input variables through relation $y = f(\boldsymbol{x}) + \varepsilon$. In a classification problem, a neural network estimates the unknown probability of output belonging to a certain class.

In a generic neural network, the input layer contains the input variables or the features in the model. Each neuron in a hidden layer receives inputs from the previous layer, executes a sequence of calculations, and passes the results on to the subsequent layer. The output layer then combines

all results to predict the outcome variable. Consider a neural network with L layers where the lth layer contains N_l neurons for $l = 1, \ldots, L$. Here layer $l = 1$ refers to the first hidden layer and layer $l = L$ refers to the output layer. In applications with a single output variable, the output layer has only one neuron, i.e. $N_L = 1$. When neuron $j(= 1, \ldots, N_l)$ in layer $l(= 1, \ldots, L)$ receives the inputs from the previous layer, denoted by $\boldsymbol{z}^{(l-1)} = \left(z_1^{(l-1)}, \ldots, z_{N_{l-1}}^{(l-1)}\right)'$, it first computes the linear combination of inputs using:

$$u_j^{(l)} = \alpha_j^{(l)} + \boldsymbol{z}^{(l-1)'}\boldsymbol{w}_j^{(l)}, \tag{1}$$

where $\boldsymbol{w}_j^{(l)} = \left(w_{j1}^{(l)}, \ldots, w_{jN_{l-1}}^{(l)}\right)'$. In the neural network literature, the intercept $\alpha_j^{(l)}$ and coefficients $\boldsymbol{w}_j^{(l)}$ are called bias and weights respectively, and they are unknown parameters to be estimated. As the notation indicates, the bias and weights are usually allowed to vary by neurons and by layers. The neuron then applies a so-called activation function to the linear index $u_j^{(l)}$ to calculate the activations:

$$z_j^{(l)} = g^{(l)}\left(u_j^{(l)}\right) \tag{2}$$

where $g^{(l)}(\cdot)$ denotes the activation function for the lth layer and it is the same for all neurons within a layer. The vector $\boldsymbol{z}^{(l)} = \left(z_1^{(l)}, \ldots, z_{n_l}^{(l)}\right)'$ forms the inputs for the $(l+1)$th layer.

In equation (1), the input variables of the model are used as the inputs for the first layer, i.e. $\boldsymbol{z}^{(0)} = \boldsymbol{x}$. The activations in the first layer $(l = 1)$ are calculated using:

$$z_j^{(1)} = g^{(1)}\left(u_j^{(1)}\right) = g^{(1)}\left(\alpha_j^{(1)} + \boldsymbol{x}'\boldsymbol{w}_j^{(1)}\right),\ j = 1, \ldots, N_1. \tag{3}$$

In the output layer $(l = L)$, the output from the neural network is calculated as:

$$\hat{y} = z_1^{(L)} = g^{(L)}\left(u_1^{(L)}\right) = g^{(L)}\left(\alpha_1^{(L)} + \boldsymbol{z}^{(L-1)'}\boldsymbol{w}_1^{(L)}\right). \tag{4}$$

In regression, $\hat{y}$ represents a point estimate for the output variable, and in classification, $\hat{y}$ represents the estimated probability of the outcome belonging to a certain class.

To train a neural network, one needs to specify a loss function which measures the discrepancy between the observed values (y) and the predicted values $(\hat{y})$ of the outcome. The standard loss functions for regression and classification problems are the mean squared error and the cross-entropy respectively. In addition, one needs to specify the functional form for the activation function in each layer. The activation function is typically chosen to be nonlinear with possible exceptions for the output layer. Nonlinearity enables the neural network to accommodate complex relations between the outcome and inputs, which is the key to the success of deep learning. Table 4 presents several candidate activation functions in the literature, among which, the rectified linear unit (ReLU) is most commonly used. The sparse representation due to zeros generated from the ReLU activation has been identified as the key element of its success (Glorot et al. (2011)). From (3) and (4), it is straightforward to see that a single layer perceptron $(L = 1)$ leads to a linear regression and a logistic regression when the linear activation function $g^{(1)}(t) = t$ and the sigmoid activation function $g^{(1)}(t) = (1 + \exp(-t))^{-1}$ are used respectively.

Table 4: Examples of activation functions

Activation	Functional Form
Linear	$g(t) = ct$
Sigmoid	$g(t) = 1/(1 + e^{-t})$
Hyperbolic tangent	$g(t) = (e^t - e^{-t})/(e^t + e^{-t})$
Rectified linear unit	$g(t) = \max\{0, t\}$

The parameters of the neural network are found to minimize the loss function. The optimization can be challenging because the error surface is non-convex, contains local minima and flat spots, and is highly multidimensional due to the large number of parameters. We emphasize two important aspects in the training process. First, the stochastic gradient descent is the general algorithm to solve the optimization where the gradient is computed via backpropagation (Rumelhart et al. (1986)). The process is iterative and in each iteration parameters are updated using only a random subset, or mini-bach, of the training data. Fitting a neural network may requires many complete passes (also known as epochs) through the entire training data. Second, training a neural network involves decisions on the hyperparameters such as the number of hidden layers, the number of neurons within a hidden layer, and the learning rate. These hyperparameters control the bias-variance trade-off in that a neural network with more neurons and more layers allows a better approximation of the unknown function, however, it is prone to overfit the unique characteristics of the training data and the predictive performance cannot be generalized to the independent test data. The rule of thumb for tuning the hyperparameters is to use cross-validation techniques to evaluate the predictive performance of the neural network for a grid of values from the hyperparameters.

4 Deep Embedding for Categorical Variables

This section details the method of categorical embedding for encoding categorical variables using neural networks. The method essentially projects categorical variables to a low-dimensional embedding space. The numeric representation, known as embdeddings, is learned in a deep neural network with a special architecture, where one uses the one-hot encodings of a categorical variable to formulate an embedding layer between the input and dense layers. The method not only applies to the rating variables in insurance such as the zip code and model and make of cars, but also can be used for any categorical variables in other fields such as Standard Occupational Classification (SOC) system, International Classification of Disease (ICD) code, or North American Industry Classification System (NAICS) code.

4.1 One-hot Encoding

The most common approach to incorporating categorical variables in statistical learning methods is to expand them into dummy variables, which is known as one-hot encoding in machine learning parlance. Consider a categorical input variable x that has K levels with values $\{c_k : k = 1, \ldots, K\}$.

One-hot encoding can be formulated as a function h that maps the categorical variable into a binary vector of length K:

$$h : x \mapsto \boldsymbol{\delta} = (\delta_{x,c_1}, \ldots, \delta_{x,c_K})', \tag{5}$$

where δ_{x,c_k}, for $k = 1, \ldots, K$, is the Kronecker delta which equals 1 if $x = c_k$ and 0 otherwise.

In principal, one-hot encoded categorical input variables are ready to use in deep neural networks. However, for a neural network to reasonably approximate any continuous function or piecewise continuous function, it requires some level of continuity in its general form and hence is not suitable to approximate arbitrary non-continuous functions (see Cybenko (1989) and Llanas et al. (2008)). Due to its continuous nature, neural networks do not favor direct use of categorical input variables, because data with categorical features may not have the minimum level of continuity or the embedded continuity is not obvious. This continuity condition is not easily met when the categorical input has a large number of levels. In addition, with one-hot encoded categorical variables, the neural network is subject to similar difficulties as in the GLMs when the cardinality of categorical variables is high. The first is the computational burden. One-hot vectors are high-dimensional and thus it requires a lot of memory to store them. The second is the estimation uncertainty. Parameter estimates are subject to high variance especially when there is not sufficient data for some levels of the categorical variable. The third is the relation between categories of the input. The encoded binary vectors for different levels are orthogonal and hence cannot reflect the similarities among them.

4.2 Categorical Embedding

Categorical embedding is an alternative method to incorporate categorical input variables in a deep learning architecture (see Guo and Berkhahn (2016)). The method maps each categorical variable into a real-valued representation in the Euclidean space and the mapping is automatically learned by a neural network in the supervised training process. In the embedding space, the categories with similar effects are close to each other, which reveals the intrinsic continuity of the categorical variable. The method of categorical embedding takes the concept behind word embedding in the natural language processing literature, where words are mapped into continuous distributed vectors in the semantic space, and similar words are identified by the distance of the embedding vectors and the direction of the difference vector (see Bengio et al. (2003) and Mikolov et al. (2013)). Because a word can be viewed as a realization from a dictionary with many entries, the methods for word embedding can be adapted to generic categorical variables.

Categorical embedding can be formulated as a function e that maps the categorical variable into a real-valued vector. For the categorical variable x with K levels, the embedding function of d-dimensional embedding space is given by:

$$e : x \mapsto \boldsymbol{\Gamma} \times \boldsymbol{\delta}, \tag{6}$$

where $\boldsymbol{\delta}$ is the one-hot encoded vector and $\boldsymbol{\Gamma} \in \mathbb{R}^{d \times K}$ is known as the embedding matrix. The

matrix $\mathbf{\Gamma}$ can be viewed as a d-dimensional numerical representation of the categorical variable and the kth column of $\mathbf{\Gamma}$ represents the kth category of x for $k = 1, \ldots, K$. To see this, for the ith data point with $x_i = c_k$, we note:

$$e(x_i) = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1K} \\ \vdots & \ddots & \vdots \\ \gamma_{d1} & \cdots & \gamma_{dK} \end{pmatrix} \times \begin{pmatrix} \delta_{x_i,c_1} \\ \vdots \\ \delta_{x_i,c_K} \end{pmatrix} = \begin{pmatrix} \gamma_{1k} \\ \vdots \\ \gamma_{dk} \end{pmatrix}. \tag{7}$$

The dimension of embedding space is bounded between 1 and $K-1$, i.e. $1 \leq d \leq K-1$. For categorical variables with a large number of levels, the dimension of embedding space is typically much smaller than the number of categories, leading to a dimension reduction.

There are $d \times K$ parameters in the embedding matrix and they can be learned during the training process for a deep learning model. To do this, we add an embedding layer, an extra layer between the input layer and the hidden layer, in the neural network architecture. The embedding layer is built with d neurons so as to map a categorical input with K levels into a d-dimensional embedding space. Figure 3 provides an conceptual illustration of the embedding layer for a single categorical variable.

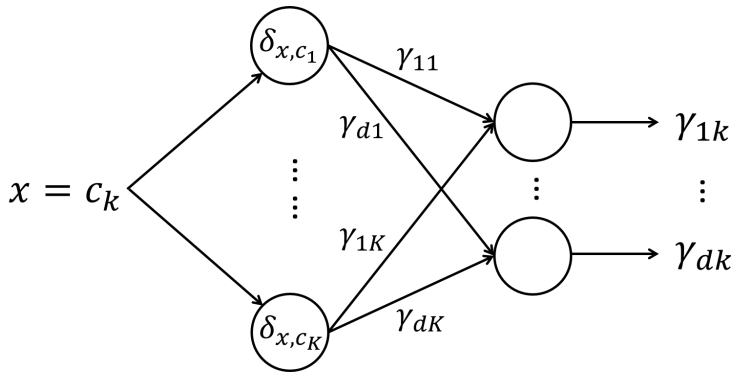

Figure 3: Graphical illustration of the embedding layer for a single categorical variable.

The embedding neurons use the one-hot encoded categorical variable as inputs to calculate the linear index and then apply an identity activation function to compute the activations. Similar to (1) and (2), the neurons in the embedding layer compute:

$$u_j^{(E)} = \boldsymbol{\delta}' \boldsymbol{w}_j^{(E)}, \tag{8}$$

$$z_j^{(E)} = g^{(E)}\left(u_j^{(E)}\right) = u_j^{(E)}, \tag{9}$$

for $j = 1, \ldots, d$, and $\boldsymbol{w}_j^{(E)} = (\gamma_{j1}, \ldots, \gamma_{jK})'$. This suggests that the embedding matrix can be viewed as the weights in the embedding layer in the neural network.

Let $\boldsymbol{z}^{(E)} = \left(z_1^{(E)}, \ldots, z_d^{(E)}\right)'$ denote the activations from the embedding layer. The embeddings of the categorical input variable and the continuous input variables are then concatenated. The merged layer is treated as a normal input layer in the neural network, and hidden and output layers

can be further built on top of it. When there is a single categorical input variable, the activations in the merged layer can be represented by $\boldsymbol{z}^{(M)} = (\boldsymbol{z}^{(E)}, \boldsymbol{x})$, and thus (3) becomes:

$$z_j^{(1)} = g^{(1)}\left(\alpha_j^{(1)} + \boldsymbol{z}^{(M)\prime}\boldsymbol{w}_j^{(1)}\right). \tag{10}$$

When there are more than one categorical input variables, each categorical input is mapped into its own embedding space and the dimension of embedding space can be different across the multiple categorical inputs. Each categorical variable generates an embedding layer in the neural network and the embedding layers are independent with each other. All embedding layers and continuous inputs are concatenated to form a merged layer. Suppose there are q categorical input variables, the activations in the merged layer can be represented by $\boldsymbol{z}^{(M)} = \left(\boldsymbol{z}_1^{(E)}, \ldots, \boldsymbol{z}_q^{(E)}, \boldsymbol{x}\right)$. Figure 4 shows a conceptual exhibition of a neural network with embedding layers.

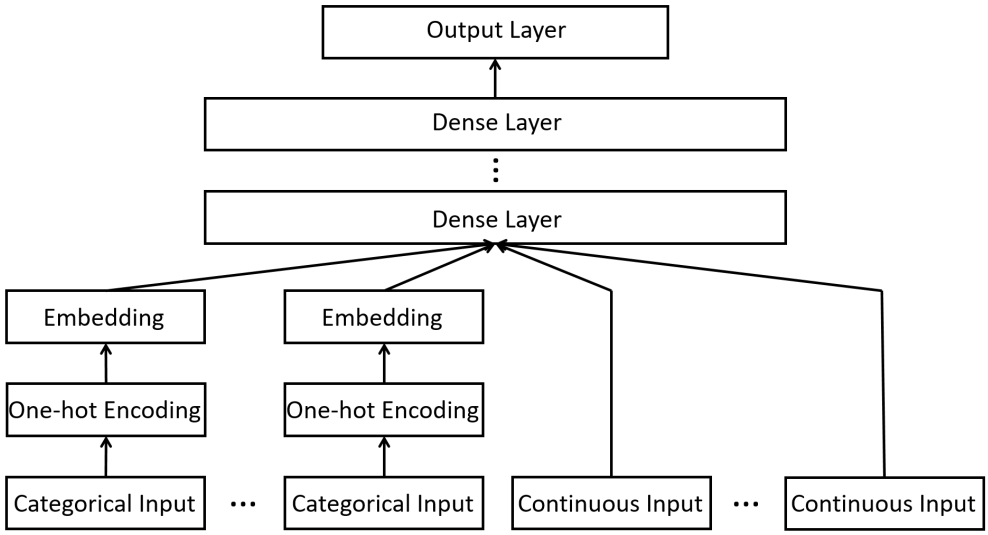

Figure 4: Conceptual exhibition of a neural network with embedding layers.

The direct connection between the weights for the embedding neurons and the embedding matrix implies that the embeddings can be estimated as part of the parameters in the neural network and no special treatment is needed. The whole network can be trained using the standard back-propagation method. The dimension of embedding space could vary across different categorical input variables and they are hyperparameters that need to be tuned in a similar way as other hyperparameters. In the training process, the embedding layer learns the intrinsic properties of each level and the relationship among different levels of the categorical variable, and the deeper layers form complex nonlinear functions of the categorical embeddings and continuous inputs.

We emphasize that categorical embedding is especially useful in two scenarios: First, in the presence of categorical variables with high cardinality, one-hot encoding generates a large number of parameters which in turn can result in overfitting. Categorical embedding, mapping the categorical variable to a low-dimensional embedding space, reduces the number of parameters substantially and generally leads to better predictions. Second, it is more often that the interest of categorical

embedding is the embedding itself rather than the predicted outcome. One important application that we demonstrate in this study is transfer learning where knowledge gathered from one task is used in another task of similar nature.

5 Actuarial Applications in Risk Classification

In this section, we apply the method of categorical embedding to the property insurance claim data described in Section 2. The outcome variable is the binary claim frequency and the categorical input of primary interest is the county code of policyholders. We demonstrate the use of categorical embedding in three different applications: The first shows how embeddings are used to construct rating classes and calculate rating relativities for a single insurance risk. The second investigates the prediction of claim frequency for multivariate insurance risks and emphasizes the effects of dependence on tail risks in portfolio claim management. The third concerns pricing new risks where data is sparse, and we showcase the novel use of transfer learning to gather knowledge from existing products. In the data analysis, we split the data into two parts. We use observations in years 2006-2011, which account for about 75% of the entire data, to train the network and to learn the embedding matrix, and we use observations in years 2012-2013 as the test set for hold-out sample comparison. The method of categorical embedding are implemented in Python, in particular, we use the deep learning framework Keras to train the model and conduct corresponding analysis.

5.1 Constructing Rating Classes for A Single Risk

Consider the context where there is a single insurance risk. Specifically, one treats the open-peril property insurance coverage as an umbrella policy and thinks of the claim frequency as a risk measurement for the aggregate claims from all peirls. To compute the embeddings for the county code, we formulate a two-class classification by a deep neural network as in Section 4.2. Specifically, we model the probability of a policyholder having at least one claims of any peril over the year. We use ReLu activation functions for the hidden layers and a Sigmoid activation function for the output layer. The dimension of the embedding space in the trained network is set to be two. Note that one should treat the embedding dimension as a tuning parameter, similar to the number of hidden layers/neurons in a neural net. We didn't observe substantial difference for a larger dimension in the exploratory analysis. In addition, we prefer a model that is as parsimonious as possible especially given the relative small data. Cross entropy is used as the loss function for early stopping purposes.

As discussed above, categorical embedding is especially valuable for prediction purposes when there is a large number of levels in the categorical variable. To illustrate, we compare the performance of two models, one with one-hot encoding and the other with categorical embedding for the county code. First, we compare the estimated probability of claims in Figure 5, where the left panel shows the fitted value from the training data and the right panel shows the predicted value

from the validation data. Although the estimated probabilities from the two networks are highly correlated, their differences are distinctive.

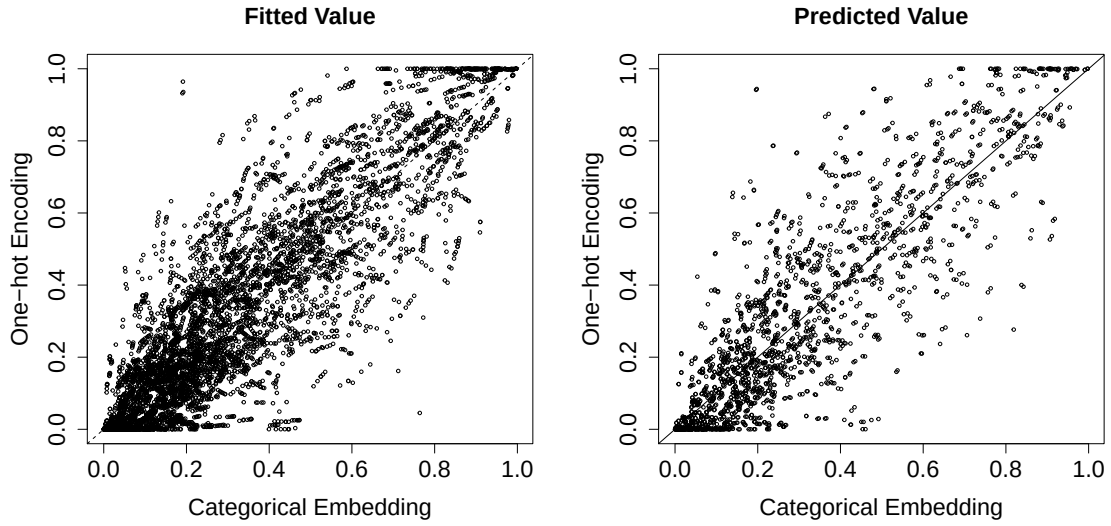

Figure 5: Comparison of one-hot encoding and categorical embedding methods. The left panel shows the fitted value from the training data and the right panel shows the predicted value from the validation data.

Second, we display the receiver operating characteristic (ROC) curves for the training and validation data in Figure 6. The left panel shows the curves from the one-hot encoding model and the right panel shows the curves from the categorical embedding model. The corresponding AUCs (area under the ROC curve) are reported in Table 5. For both methods, the AUC for training data is larger than that for test data, which is not surprising given that one is generalizing the prediction performance to a new data set. However, the difference in AUCs for the training and the validation data from the one-hot encoding model is much higher than the difference from the categorical embedding model. For one-hot encoding, the AUCs are 87.7% and 78.5%, and for categorical embedding, the AUCs are 82.7% and 81.0%, for the training and the test data, respectively. This distinctive difference is consistent with the theoretical implication that one-hot encoding tends to overfit the data due to the large number of levels in the county code. Despite the worse goodness-of-fit of the categorical embedding model, it outperforms the one-hot encoding method in the validation set.

Table 5: AUCs for training and validation data from alternative models.

	Training Set	Validation Set
One-hot Encoding	0.877	0.785
Categorical Embedding	0.827	0.810

As another comparison, we resort to the Gini index proposed by Frees et al. (2011). We report

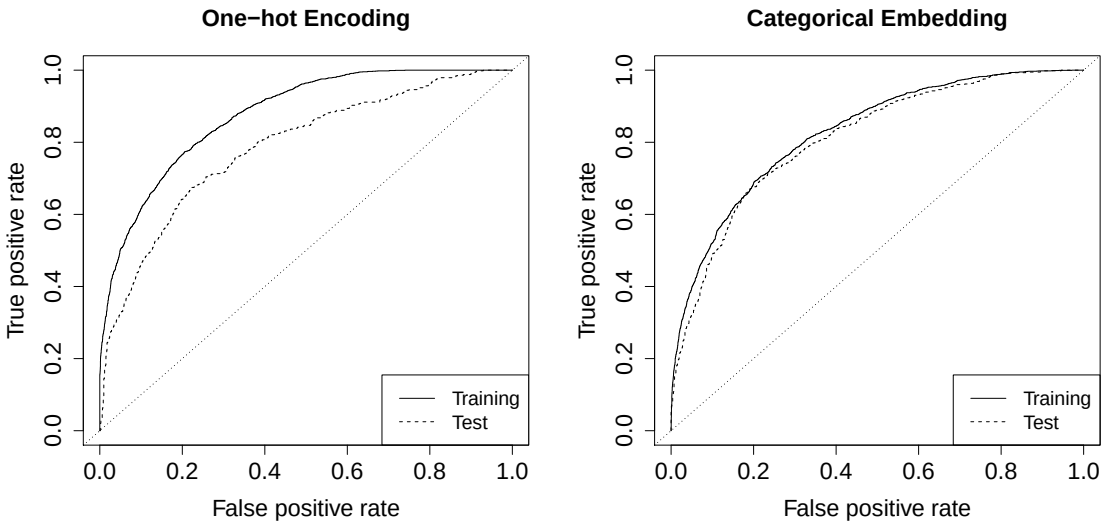

Figure 6: ROC curves for training and validation sets. The left panel shows the curves from the one-hot encoding model and the right panel shows the curves from the categorical embedding model.

in Table 6 the simple Gini with the associated standard error for the two alternative methods. The simple Gini is calculated using the validation data and a larger Gini index indicates better performance. The smaller standard error suggests that the categorical embedding model has significant higher Gini index, and thus superior predictive performance than the one-hot encoding model. We further perform a champion-challenger test between the predictions from the two methods and the results are also reported in Table 6. In this test, we use one prediction as base rate and the other as alternative rate, and we examine whether the insurer could identify additional profit opportunities when switching from the base to the alternative. When the prediction from one-hot encoding method is used as the base, the Gini index is 21.152 with a standard error of 2.050. The large statistic rejects the base and suggests that the categorical embedding method improves the separation between good and bad risks. In contrast, when the prediction from the categorical embedding method is used as the base, the small and insignificant Gini index implies that additional profit opportunities are not obvious if the insurer looks to the alternative prediction.

Table 6: Prediction comparison using Gini statistics[†]

	One-hot Encoding	Categorical Embedding
Simple Gini	41.534 (1.745)	45.204 (1.619)
One-hot Encoding		21.152 (2.050)
Categorical Embedding	3.049 (2.004)	

[†] Standard errors are reported in parentheses.

The analysis thus far has focused on the predictive aspect of the categorical embedding method.

As we stressed in Section 4.2, the more appealing output from the embedding method is the embeddings of the categorical variable instead of the prediction. Suppose the task of the insurer is territorial risk classification, i.e. to establish geographical rating classes in the risk classification system and calculate the associated relativity for each risk class (Shi et al. (2017)). Existing geographical regions such as those defined by postal code or municipality boundaries might be coarse and need to be refined. We show that embeddings can be employed to create risk classes in this application.

Recall that embeddings are numerical representation of categorial variables. Specifically, the county code has 72 levels and the trained dimension of the embedding space is 2, therefore the embeddings can be represented by a 72×2 matrix. Each row corresponds to one county, and the rows should be close for similar counties. We perform a clustering analysis and identify five distinctive clusters. Figure 7 displays the five risk classes on both embedding space and principal component space where clusters are ordered from low risk to high risk. The convex boundary completely separates the five subgroups in both plots, which suggests that the counties are reasonably grouped given the relatively small data. As another visualization, we show the rating classes on a heat map as in Figure 8. It is interesting to observe some spatial continuity, i.e. a county tends to cluster with its neighboring counties, despite the fact that the training process does not take into account any spatial information. The results can be explained by the unobserved spatial heterogeneity and suggest that the method can be used to identify underlying spatial subgroups.

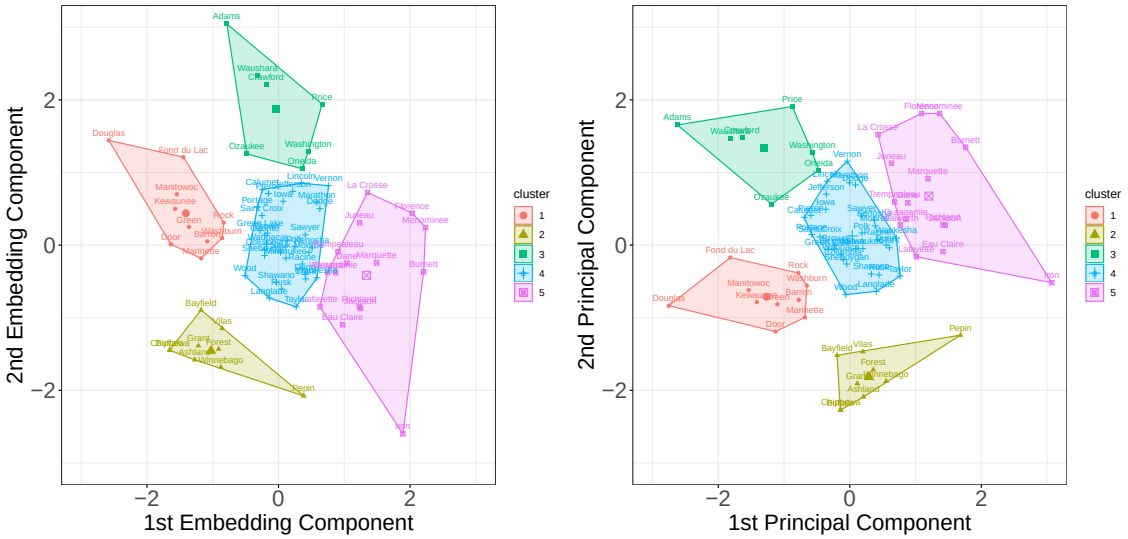

Figure 7: Representation of distinctive risk classes on embedding space (left) and principal component space (right).

We calculate the relativity for each risk class using logistic regressions. See Werner and Modlin (2016) for the connection between the GLMs and the traditional risk classification methods. The relativity indicates the risk of a given class relative to the base that is prespecified. Figure 9 displays

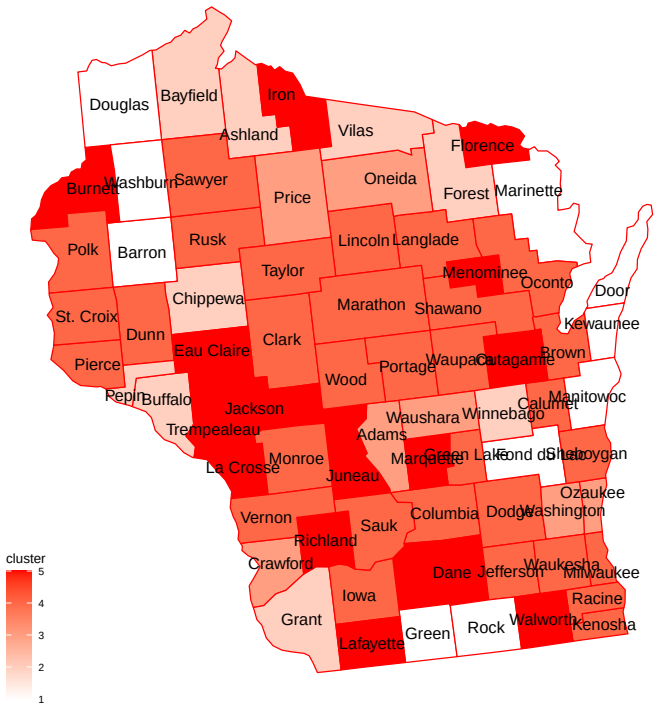

Figure 8: Heat map of rating classes by county of Wisconsin.

the relativity along with the confidence band from the univariate and multivariate methods. The difference is that the multivariate method takes into account the potential confounding effect of other rating variables while the univariate method does not. For this particular dataset, ignoring the correlation among rating variables substantially underestimate the relativity for territorial classes. The significant difference from the implied relativity to the base class supports the segmentation of risks by the refined territorial clusters. The large uncertainty in the relativity is due to the small sample size, especially for the middle risk class. Finally, we compare the goodness-of-fit of two logistic regressions, one using the original county code and the other using the refined county code as the territorial rating variable. The AIC and BIC statistics are 8,480 and 9,041 for the former, and 8,404 and 8,489 for the latter, which reinforces the fact that the dimension-reduction due to refined territorial classes help avoid the potential overfitting.

5.2 Portfolio Management for Dependent Risks

Short-term insurance contracts are often featured with a "bundling" design. For instance, a comprehensive automobile insurance policy provides coverage for both collision and third-party liability; a worker's compensation insurance provides benefits for wage replacement, medical treatment, and vocational rehabilitation; an-open peril property insurance policy covers losses due to all types of causes subject to certain exclusions. The bundling products involve multiple insurance risks which

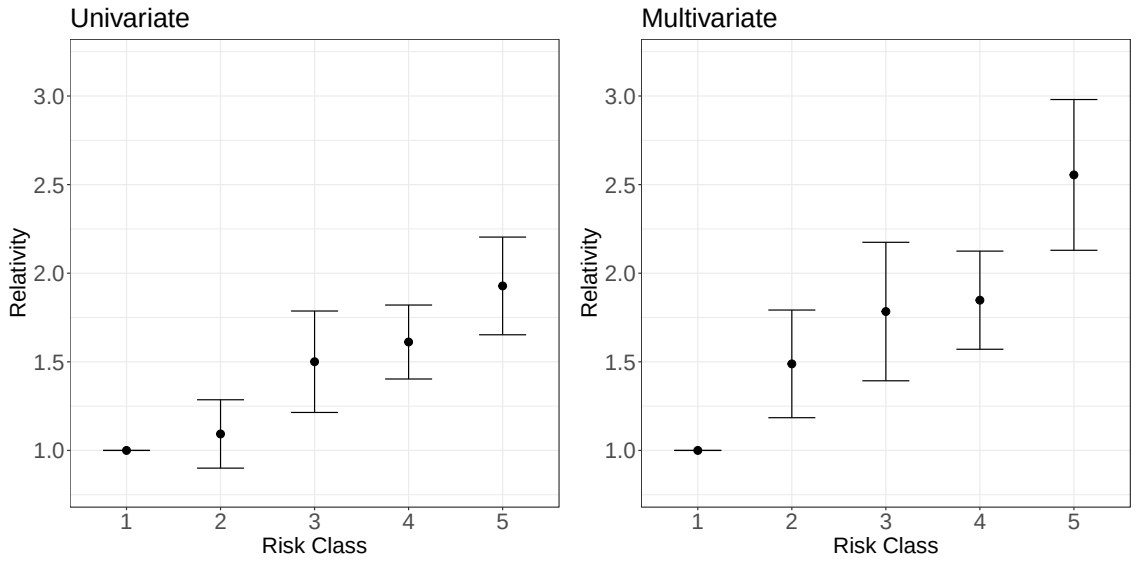

Figure 9: Relativity of risk classes using univariate and multivariate methods.

tend to be correlated with each other. In the case of dependent insurance risks, it is appealing to consider a joint modeling framework to account for such dependency, for instance, see Frees et al. (2009) and Shi et al. (2016) for automobile insurance, Frees et al. (2010) and Yang and Shi (2019) for property insurance, and Frees et al. (2013) for health insurance.

In this study, the claim frequency is measured for fire, water, and other perils for each policyholder. Instead of examining the aggregate risk from all perils, we look into the claim frequency by peril. Let Z_j, $j = 1, 2, 3$, denote the binary claim indicator for the three perils. If the three outcomes are independent with each other, one could directly apply the neural network to each outcome separately as in Section 5.1. However, the exploratory analysis in Section 2 indicates dependence among the three perils and suggests some joint modeling strategy.

The goal of this application is to build a deep learning structure for the prediction of claim frequencies of dependent risks and to incorporate categorical inputs into this neural network using categorical embedding. To accommodate the association among the multiple perils in a neural network, we define a new output variable $Y = (Z_1, Z_2, Z_3)$ and model Y as a categorical variable with eight levels. The labels and observed frequency for Y from the training data are summarized in Table 7. In addition, we also report in the table the association ratio for each level which is defined as:

$$\rho(z_1, z_2, z_3) = \frac{\Pr(Z_1 = z_1, Z_2 = z_2, Z_3 = z_3)}{\Pr(Z_1 = z_1)\Pr(Z_2 = z_2)\Pr(Z_3 = z_3)} \tag{11}$$

Consistent with Table 3, the association ratio implies the positive relationship among the peril-wise claim frequency. We stress that the usual strategy for modeling dependent risks is to consider the joint distribution of Z_1, Z_2, and Z_3 (see Shi and Guszcza (2016) for more details). In a neural network, one could use an multi-output structure where the output layer contains where the

output layer contains three neurons with each corresponding to one binary outcome. However, this structure won't allow us to capture the dependence among the three binary claim outcomes. In contrast, we transform the joint modeling of (Z_1, Z_2, Z_3) to a multi-class classification problem for output Y.

The analysis in Section 5.1 has demonstrated the value of categorical embedding in the context of a single insurance risk. The merit of categorical embedding is more pronounced in case of multiple dependent risks where model complexity increases exponentially as the number of risks increases. To perform categorical embedding, we build a similar deep architecture in the neural network to the one shown in Figure 4. The only difference is that the output layer consists of eight neurons with each corresponding to a category of Y. To train the neural network, the softmax activation is used in the output layer and the output from the network becomes:

$$\hat{y}_j = g^{(L)}\left(\boldsymbol{u}^{(L)}\right) = \frac{\exp\left\{u_j^{(L)}\right\}}{\sum_{j=1}^{8} \exp\left\{u_j^{(L)}\right\}}, \quad \text{for} \quad j = 1, \ldots, 8, \tag{12}$$

where $u_j^{(L)} = \alpha_j^{(L)} + \boldsymbol{z}^{(L-1)'}\boldsymbol{w}_j^{(L)}$ and $\boldsymbol{u}^{(L)} = \left(u_1^{(L)}, \ldots, u_8^{(L)}\right)$. In expression (12), $\hat{y}_j$ represents the predicted probability that outcome Y belongs to the jth class, and the softmax activation function ensures $\sum_{j=1}^{8} \hat{y}_j = 1$. We present in Figure 10 the trained architecture of the neural network for the multivariate dependent insurance risks, and the embeddings for county code are obtained in the training process. As shown in the figure, categorical input variables are mapped to a two-dimensional numeric vector in the embedding layer, the embedding layers are then concatenated to construct the input for a dense layer. The output from the dense layer is further concatenated with the continuous input variables to form inputs for the hidden layers. The final output from the network is of dimension eight with each element representing the estimated probability of being in a given class.

Table 7: Observed and fitted frequencies of the risk outcome for dependent risks

	Observed	Association	Fitted Value	
	Value	Ratio	Independent	Dependent
$(0,0,0)$	4,745	1.450	4,687	4,757
$(0,0,1)$	410	0.830	431	401
$(0,1,0)$	444	0.849	478	453
$(1,0,0)$	557	0.963	599	550
$(0,1,1)$	102	1.264	116	101
$(1,0,1)$	157	1.810	143	152
$(1,1,0)$	178	1.862	176	179
$(1,1,1)$	165	11.597	128	164
$\chi^2 - statistic$			20.883	0.686

Given that the peril-wise claim frequency outcomes are correlated, ignoring the correlation among them will lead to poor goodness-of-fit of the model. Table 7 summarizes the fitted fre-

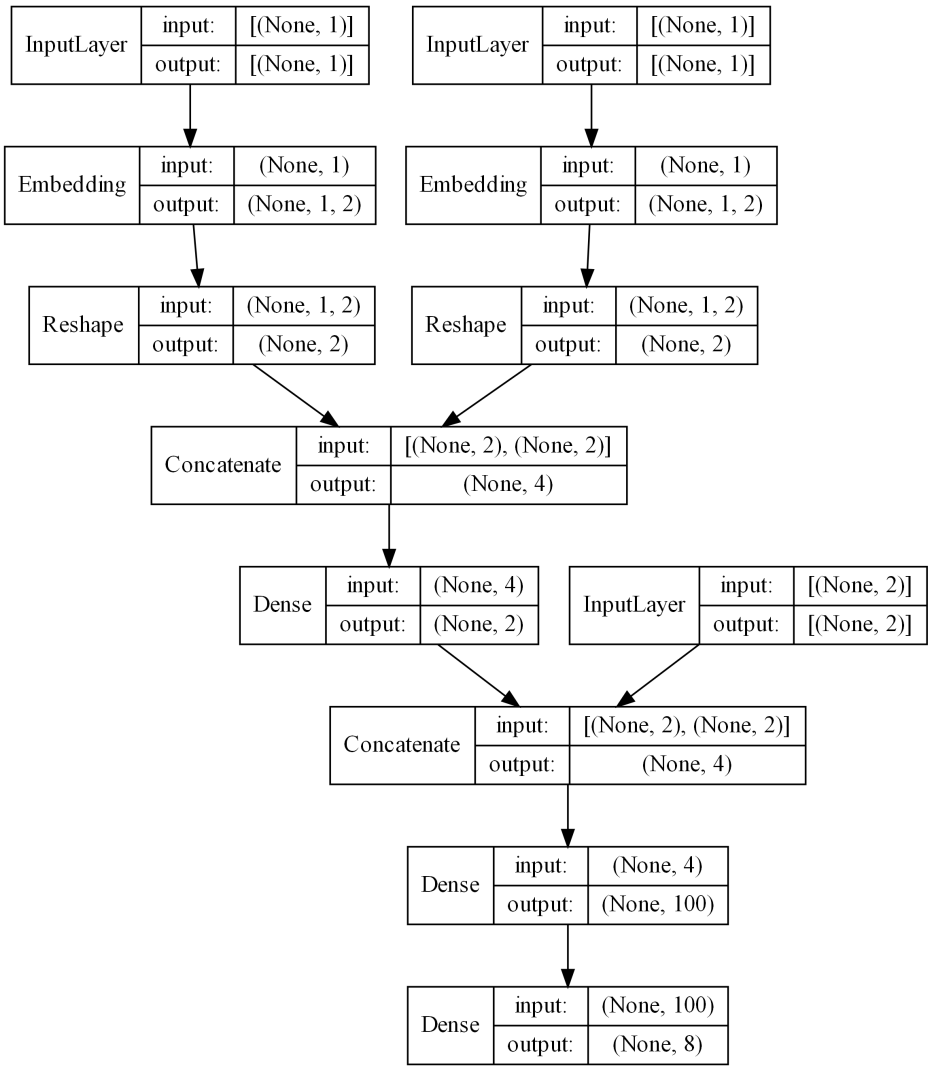

Figure 10: Deep architecture in the neural network for dependent insurance risks.

quency of Y from the independent and dependent models. For a given level of Y, the fitted value is computed by summing the estimated probability across all individuals in the training data. For categories with association ratio greater than one, the independent model underestimates the frequency, and for categories with association smaller than one, the independent model overestimates the frequency. The corresponding χ^2-statistics for the independent and dependent models are also reported in Table 7, which further confirms the favorable fit of the trained neural network.

Last, we consider the effect of dependence among multiple insurance risks on the portfolio loss distribution with a focus on the tail risks. The portfolio consists of policyholders in the hold-out sample. For each policyholder in the portfolio, define the total loss cost as $S = c_1Z_1 + c_2Z_2 + c_3Z_3$. One can think of c_j, $j = 1, 2, 3$, as the claim amount for peril j, which can be either fixed or random. We consider two types of insurance coverage, the stop-loss insurance and the excess-of-loss insurance. The insurer's retained loss can be represented as:

$$\text{Stop loss} : R_1 = \min\{S, d_1\}$$
$$\text{Excess of loss} : R_2 = \max\{S - d_2, 0\}$$

The stop-loss and excess-of-loss coverage focus on the left and right tail of aggregate loss S respectively. In the numerical experiment, we set $c_1 = c_2 = c_3 = 1$, and $d_1 = 0.5$ and $d_2 = 1.5$. The insights gained from this simple setting readily extend to the generic setting. Assuming the stop-loss or the excess-of-loss coverage applies to each individual in the portfolio, we display in Figure 11 the distribution of the insurer's total retained losses for the portfolio. The left panel shows the distribution for stop-loss insurance and the right panel shows the distribution for excess-of-loss insurance. For each scenario, we generate the loss distribution from both independent and dependent models so as to examine the effect of dependence among multiple insurance risks. The independent model significantly overestimates the portfolio loss for the stop-loss coverage, and underestimates the portfolio loss for the excess-of-loss coverage. In summary, when multiple insurance risks are positively (negatively) correlated, ignoring the dependence will overprice (underprice) the lower-tail risk, and underprice (overprice) the lower-tail risk in the insurance portfolio.

5.3 Pricing New Risks

Risk classification usually requires insurers to have access to a large amount of historical claims data. One challenge that insurers often face when pricing new insurance coverage is the sparsity of data. We show that the categorical embedding method can be particularly useful in risk classification for new insurance coverage. Recall that the property fund data contains claim frequency from three perils, fire, water, and other. To illustrate the idea, suppose that the insurer has only provided coverage for water and other perils during years 2006-2011. Starting from year 2012, the insurer plans to offer fire coverage as well. That is, the insurer's database (training data) contains claim experiences of policyholders for water and other perils, but not fire peril. The task is to establish a risk classification system for the insurer to underwrite and price the fire coverage.

In this application, we focus on the geographical region and aim to create territorial risk classes

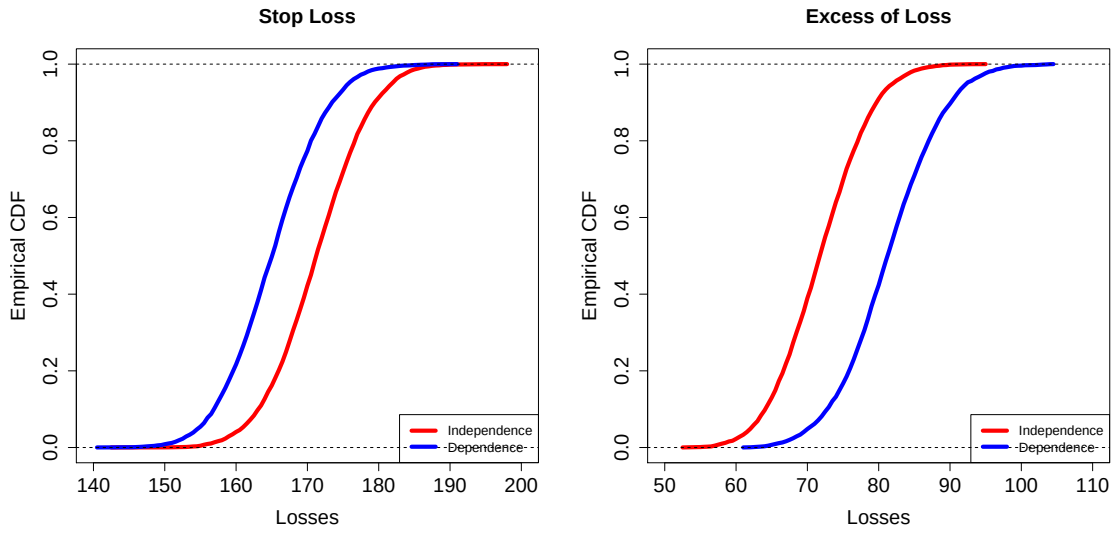

Figure 11: Portfolio loss distribution. The left panel shows the distribution for stop-loss insurance and the right panel shows the distribution for excess-of-loss insurance.

from the county code of policyholders. In doing so, we employ the method of transfer learning where knowledge gathered from one task is used in another task of similar nature. Transfer learning has a long history in machine learning and the advent of deep learning has led to a range of new transfer learning approaches (see surveys by Pan and Yang (2009) and Tan et al. (2018)). The essential idea is that although the insurer doesn't have any or not enough loss experience on fire peril, the insurer could potentially construct territorial risk classes for the fire peril using loss experience on water and other perils. Intuitively, the dependence among multiple peril risks motivates the transfer learning in this context. To emphasize, the prior information required is the conceptual knowledge that a high-risk policyholder in one peril is more likely to be high risk in other perils. It does not require one to quantify such information, but such assumptions must be known to justify the use of transfer learning.

We consider two strategies to demonstrate the idea. In the first one, we learn the territorial risk classes for fire peril from a single related peril, be it water or other. We apply the deep neural network for a single insurance risk in Section 5.1 to the claim frequency from water and other perils separately to train the embeddings for county code. In the second one, we learn the territorial risk classes for fire peril from both water and other perils simultaneously. That is, the embedding matrix for county code is trained using the deep neural network for dependent risks developed in Section 5.2. In all scenarios, we set the dimension of embedding space equal to two. As a result, we obtain three 72×2 embedding matrices of county code, one learned from the water peril, one learned from the other peril, and the last one learned from water and other perils jointly.

The embedding matrix provides insights on the relationship among the counties in terms of their effects on claim frequency. One expects that the embeddings of similar counties are close to each

other. To visualize the similarity among counties, we exhibit in Figure 12 the similarity matrix that describes the pair-wise closeness among the 72 counties. The figure shows the similarity matrix from three cases. The plots in left panel from top to bottom are based on the embedding matrix learned from water peril, other peril, and both perils jointly, respectively. Using such analysis, the insurer could obtain preliminary knowledge on the latent subgroups within the county code. In all three cases, the similarity matrix indicates some clustering effects among counties. As a more formal strategy, the insurer could perform a clustering analysis to identify the subgroups. We group the embeddings into three rating classes which are further plotted on the map of Wisconsin in the right panel of Figure 12. The rating classes in the right panel are consistent with the ordering of counties in the left panel. Recall that our ultimate goal is to learn territorial risk classes for fire peril, and because of lack of data, we gather information from other perils. Specifically, in univariate case, we perform learning using data from water peril and other peril separately, and in bivariate case, we perform learning using data from water and other perils simultaneously. Compared across the three cases, the results in Figure 12 suggests noticeable differences. This is not surprising given that the learning processes are supervised by three different output variables. In the univariate case, it is the marginal distribution of claim frequency from water peril or other peril that supervises the learning of embeddings, and in the bivariate case, it is the joint distribution of claim frequency from water and other perils that directs the learning. The result from the bivariate case appears to be balance between the two univariate cases. Despite the difference, one expects that the embedding matrices learned from water and other perils to be somewhat informative to the peril of fire, because of the dependence of fire peril risk with both water and other perils.

When pricing new insurance coverage or products, an insurer typically starts with industry experience and continues to refine the risk classification system as more data are collected. We examine whether the embeddings of county code learned from water and other perils are predictive for fire peril using the fire peril claim data in year 2012 in the validation set. For illustration, we fit a logistic regression using binary claim frequency of fire peril as the response. The embeddings of county code obtained using transfer learning are used as predictors in two ways. In the first approach, one directly uses the learning embedding vectors, and in the second approach, one uses the territorial risk classes generated from the embeddings. The former is more suitable for underwriting practice where the insurer is more interested in the risk score of policyholders. In contrast, the latter is more relevant to the ratemaking where rates are often quoted for homogenous risk classes. In addition, the model also controls for the other rating variables including entity type, coverage amount, and deductible. Table 8 summarizes the estimated parameters for models using embeddings learned from water, other, and both perils. First, the embedding covariates based on both embedding clusters and embedding matrices show a significant effect on claim frequency in all models, indicating the effectiveness of the transfer learning. Second, the implied relativity of embedding covariates varies across different transfer learning models, which is explained by the fact that the embeddings of county code are learned in neural networks supervised by different outcomes. However, the goodness-of-fit of all models in Table 8 are comparable based on the value

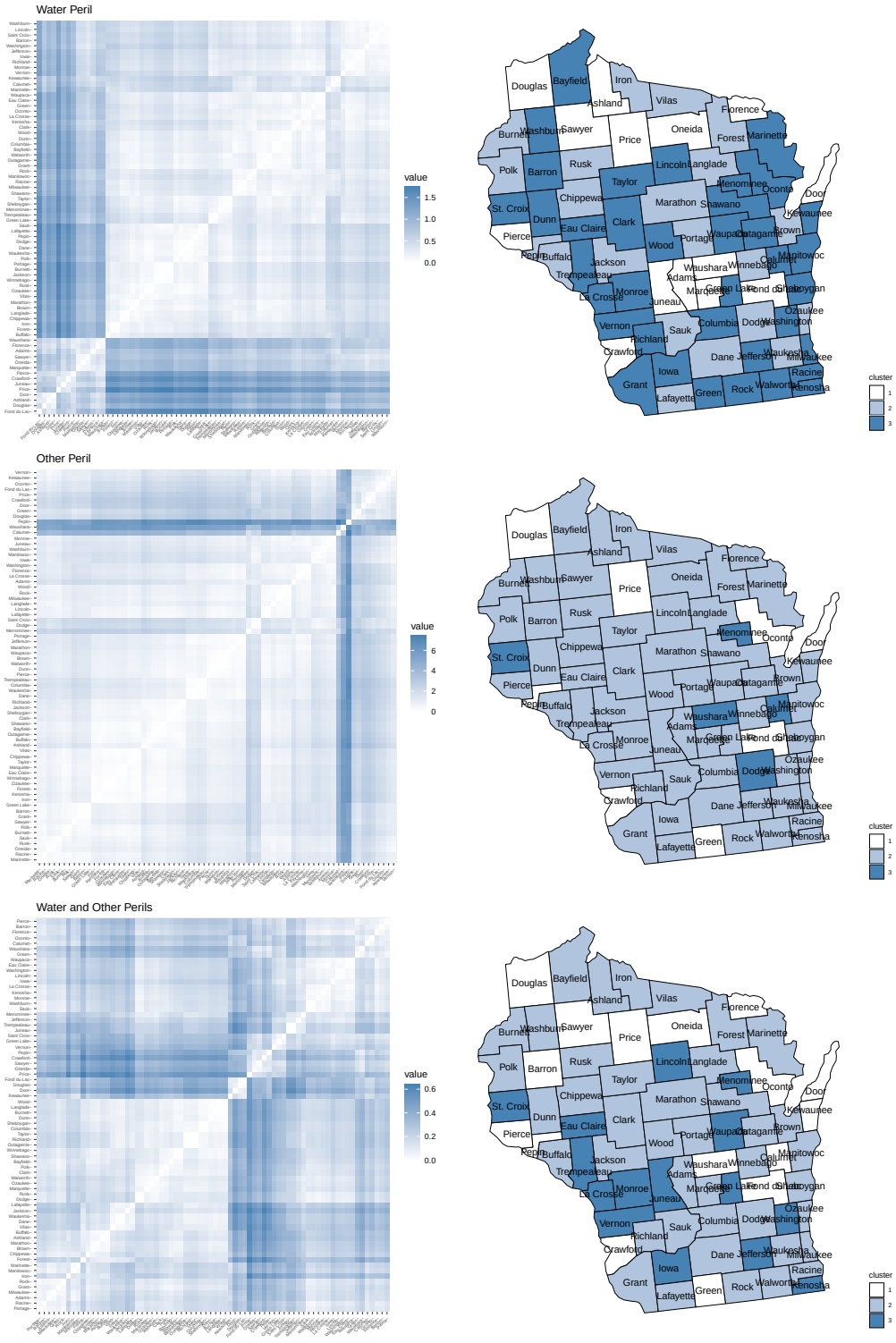

Figure 12: Similarity matrix and clusters of counties from categorical embedding. From top to bottom, the plots in a row correspond to the case learned from the water peril, other peril, and water and other perils, respectively.

of the log-likelihood function reported along with each model.

We emphasize that one purpose of categorical embedding is to avoid overfitting via dimension reduction. Instead of embeddings obtained using transfer learning, the insurer could directly use county code for pricing the new peril of fire. However, it could lead to overfitting in two scenarios of subtle differences: first, there exists subgroups among counties that have different effects on claim frequency; second, the data is sparse regardless of presence of subgroups. Using either embedding clusters or embedding matrix, we reduce the dimension of county code from the number of levels to the dimension of embedding space, which is expected to prevent overfitting. To verify this intuition, we use the claims data from fire peril in year 2013 in the validation set to test this hypothesis.

First, we refer to the ROC curve using the test data. The ROC curves are compared between models with and without transfer learning in Figure 13. The left panel uses embeddings learned from either water peril or other peril separately, and the right panel uses embeddings jointly learned from water and other perils. The corresponding AUCs are 77% and 81% for models without and with transfer learning, respectively. The improvement in the AUC suggests that the direct use of county code leads to overfitting in the fire claim frequency model. Using categorical embedding could help address this issue, even when the embeddings are not directly learned from the fire peril but indirectly from dependent perils. In addition, our analysis show that the difference among AUCs from different transfer learning models are not apparent, which is in line with the goodness-of-fit statistics in Table 8.

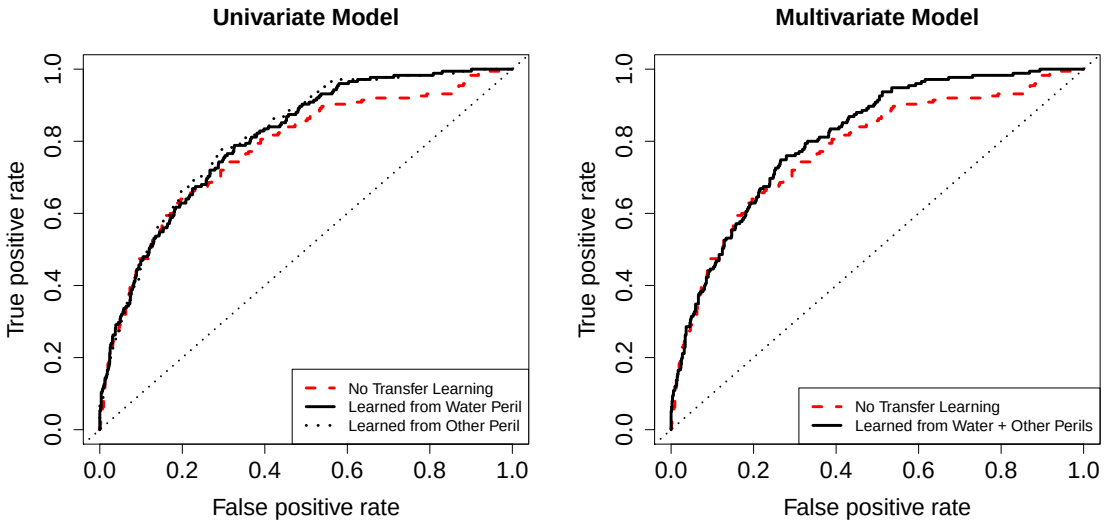

Figure 13: Comparison of ROC curves between models with and without transfer learning. The left panel uses embeddings learned from a single risk, and the right panel uses embeddings jointly learned from two risks.

Second, we perform an alternative test using the Gini index as an additional support for transfer learning. To show the effectiveness of transfer learning, we compare the predictions of claim

Table 8: Estimated logistic regression for fire peril using transfer learning

	Based on Embedding Cluster				Based on Embedding Matrix		
	Estimate	Std. Error	p-value		Estimate	Std. Error	p-value
Learned from Water Peril							
(Intercept)	-2.557	0.855	0.003	(Intercept)	-1.439	0.797	0.071
City	1.327	0.525	0.012	City	1.281	0.525	0.015
County	1.495	0.565	0.008	County	1.357	0.563	0.016
School	-0.006	0.527	0.991	School	-0.046	0.526	0.930
Town	1.328	0.654	0.042	Town	1.357	0.653	0.038
Village	1.345	0.530	0.011	Village	1.330	0.528	0.012
Coverage	0.889	0.109	0.000	Coverage	0.926	0.111	0.000
Deductible	-0.485	0.095	0.000	Deductible	-0.482	0.094	0.000
Region2	0.982	0.414	0.018	Embedding1	1.430	0.744	0.055
Region3	1.120	0.399	0.005	Embedding2	-0.389	0.283	0.169
χ^2-statistic		6.029		χ^2-statistic		6.150	
Loglik		-371.079		Loglik		-372.769	
Learned from Other Peril							
(Intercept)	-2.614	0.897	0.004	(Intercept)	-1.239	0.810	0.126
City	1.298	0.525	0.013	City	1.280	0.522	0.014
County	1.402	0.562	0.013	County	1.448	0.561	0.010
School	-0.017	0.527	0.974	School	-0.026	0.524	0.960
Town	1.373	0.654	0.036	Town	1.259	0.650	0.053
Village	1.357	0.530	0.011	Village	1.311	0.527	0.013
Coverage	0.919	0.109	0.000	Coverage	0.893	0.109	0.000
Deductible	-0.489	0.094	0.000	Deductible	-0.499	0.095	0.000
Region2	1.025	0.484	0.034	Embedding1	1.606	0.685	0.019
Region3	1.279	0.605	0.034	Embedding2	-0.141	0.305	0.643
χ^2-statistic		9.530		χ^2-statistic		6.246	
Loglik		-372.830		Loglik		-372.721	
Learned from Water and Other Perils							
(Intercept)	-2.095	0.818	0.010	(Intercept)	-1.188	0.813	0.144
City	1.280	0.524	0.015	City	1.212	0.524	0.021
County	1.377	0.564	0.015	County	1.295	0.562	0.021
School	-0.048	0.527	0.927	School	-0.107	0.526	0.840
Town	1.342	0.651	0.039	Town	1.329	0.653	0.042
Village	1.316	0.529	0.013	Village	1.294	0.528	0.014
Coverage	0.909	0.110	0.000	Coverage	0.940	0.112	<2e-16
Deductible	-0.483	0.095	0.000	Deductible	-0.485	0.095	0.000
Region2	0.466	0.312	0.135	Embedding1	2.658	1.039	0.011
Region3	0.825	0.346	0.017	Embedding2	-1.175	0.771	0.128
χ^2-statistic		6.139		χ^2-statistic		8.306	
Loglik		-372.775		Loglik		-371.691	

frequency of fire peril obtained with and without using transfer learning. In the case of transfer learning, predictions are computed using each of the six models presented in Table 8. Without transfer learning, we consider two scenarios: the base model charges average cost across all counties, and the refined model charges county-specific cost. We use the rate based on county embeddings to challenge both the base and refined rate. Gini indices are computed using test data in year 2013 and the results are presented in Table 9. Consistent with the estimates from Table 8, the Gini indices are comparable across different transfer learning predictions, and the large statistics suggest that the rating plan could be further refined with the embeddings obtained from transfer learning.

Table 9: † Comparison of predictions from transfer learning using Gini index

	Embedding Cluster			Embedding Matrix		
	Water	Other	Water+Other	Water	Other	Water+Other
Average	50.973	51.502	51.280	51.068	52.162	51.725
	(2.906)	(2.887)	(2.878)	(2.887)	(2.829)	(2.851)
County	20.892	21.393	22.640	20.504	23.622	22.496
	(4.029)	(3.981)	(3.931)	(4.044)	(3.869)	(3.904)

† Standard errors are reported in parentheses.

Last, we perform an analysis to compare performance of the networks with transfer learning to the case where the 72 counties are arbitrarily grouped. The former case consists of two models, one uses 3 groups of counties obtained from clustering the embeddings as above, and the other uses the trained embedding vector as predictors. In the latter case, we randomly assign counties into 3 groups (to be consistent with the number of groups we obtained based on embeddings). We compare in-sample and out-of-sample performances using AIC and AUC respectively. We replicate the experiment 500 times. When the embeddings for counties are learned from water and other perils jointly, 94% and 97% of times the two transfer learning models (one uses embedded clusters and the other uses embedding vectors) have higher AIC respectively, and 63% and 95% of times the two transfer learning models have higher AUC respectively.

In conclusion, we emphasize that pricing new coverage or products could be challenging because of the sparsity of data and this process could become even more complicated when categorical rating variables have a large number of levels. Categorical embeddings learned from related risks, although not necessarily perfect, improve the insurer's decision making in risk classification.

6 Conclusion

We introduced the method of embedding categorical variables as a tool for risk classification for nonlife insurance products. The method was mainly motivated by the challenges that traditional actuarial models have with categorical rating variables, especially those with a large number of levels. Our work was problem driven in that we demonstrate novel applications of deep embedding

method in three actuarial applications. We first showed how embeddings are used in risk classification for a single insurance risks. Then we examined the predictive model for multivariate dependent insurance risks. Last, we demonstrated using categorical embedding for pricing new risks through transfer learning.

The three applications demonstrated and emphasized two distinctive aspects of the proposed categorical embedding method. First, because the method is formulated as an artificial neural network and is automatically trained in the supervised learning process, it is natural to view it as a mechanism to incorporate categorical input variables in deep neural networks. This perspective emphasized the predictive aspect of the method and is illustrated by predictive applications for both univariate and multivariate insurance risks. Second, the neural network is viewed as a vehicle for computing the embeddings for categorical input variables. The embeddings themselves are the primary interests instead of prediction, and output variable simply serves as the supervisor who directs the learning of embeddings. This perspective has been demonstrated in particular in the transfer learning applications.

In addition to nonlife insurance risk classification, many other applications in the area of insurance analytics involve categorical variables of high cardinality. We anticipate that the method of categorical embedding and the two distinctive aspects identified in this work will receive more attention and help improve decision making in insurance company operations.

References

Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin (2003). A neural probabilistic language model. *Journal of Machine Learning Research 3*, 1137–1155.

Brown, R. L. (1988). Minimum bias with generalized linear models. In *Proceedings of the Casualty Actuarial Society*, Volume 75, pp. 187–217. Citeseer.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems 2*(4), 303–314.

De Jong, P., G. Z. Heller, et al. (2008). Generalized linear models for insurance data.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186.

Frees, E. W., X. Jin, and X. Lin (2013). Actuarial applications of multivariate two-part regression models. *Annals of Actuarial Science 7*(2), 258–287.

Frees, E. W. and G. Lee (2015). Rating endorsements using generalized linear models. *Variance 10*(1), 51–74.

Frees, E. W., G. Meyers, and A. D. Cummings (2011). Summarizing insurance scores using a gini index. *Journal of the American Statistical Association 106*(495), 1085–1098.

Frees, E. W., P. Shi, and E. A. Valdez (2009). Actuarial applications of a hierarchical insurance claims model. *ASTIN Bulletin: The Journal of the IAA 39*(1), 165–197.

Frees, E. W. J., G. Meyers, and A. D. Cummings (2010). Dependent multi-peril ratemaking models. *ASTIN Bulletin: The Journal of the IAA 40*(2), 699–726.

Glorot, X., A. Bordes, and Y. Bengio (2011). Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pp. 315–323.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. Cambridge: MIT Press.

Guo, C. and F. Berkhahn (2016). Entity embeddings of categorical variables. *arXiv:1604.06737*.

Haberman, S. and A. E. Renshaw (1996). Generalized linear models and actuarial science. *Journal of the Royal Statistical Society: Series D (The Statistician) 45*(4), 407–436.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pp. 1746–1751.

Klinker, F. (2011). Generalized linear mixed models for ratemaking: a means of introducing credibility into a generalized linear model setting. *Casualty Actuarial Society E-Forum 2*, 1–25.

Kuo, K. and R. Richman (2021). Embeddings and attention in predictive modeling. *arXiv preprint arXiv:2104.03545*.

Lai, S., L. Xu, K. Liu, and J. Zhao (2015). Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI Conference on Artificial Intelligence*, pp. 2267–2273.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *Nature 521*(7553), 436–444.

Lee, G. Y., S. Manski, and T. Maiti (2020). Actuarial applications of word embedding models. *ASTIN Bulletin: The Journal of the IAA 50*(1), 1–24.

Llanas, B., S. Lantarón, and F. J. Sáinz (2008). Constructive approximation of discontinuous functions by neural networks. *Neural Processing Letters 27*(3), 209–226.

McCulloch, W. S. and W. Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 5*(4), 115–133.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Mildenhall, S. J. (1999). A systematic relationship between minimum bias and generalized linear models. In *Proceedings of the Casualty Actuarial Society*, Volume 86, pp. 393–487.

Ohlsson, E. and B. Johansson (2006). Exact credibility and tweedie models. *ASTIN Bulletin: The Journal of the IAA 36*(1), 121–133.

Pan, S. J. and Q. Yang (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering 22*(10), 1345–1359.

Perla, F., R. Richman, S. Scognamiglio, and M. V. Wüthrich (2021). Time-series forecasting of mortality rates using deep learning. *Scandinavian Actuarial Journal*, 1–27.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *nature 323*(6088), 533–536.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks 61*, 85–117.

Shi, P., X. Feng, J.-P. Boucher, et al. (2016). Multilevel modeling of insurance claims using copulas. *The Annals of Applied Statistics 10*(2), 834–863.

Shi, P. and J. Guszcza (2016). Frameworks for general insurance ratemaking: Beyond the generalized linear model. In E. Edward, G. Meyers, and R. A. Derrig (Eds.), *Predictive Modeling Applications in Actuarial Science: Volume II, Case Studies in Insurance*, Cambridge, pp. 100–125. Cambridge University Press.

Shi, P., K. Shi, et al. (2017). Territorial risk classification using spatially dependent frequency-severity models. *Astin Bulletin 47*(2), 437–465.

Shi, P. and L. Yang (2018). Pair copula constructions for insurance experience rating. *Journal of the American Statistical Association 113*(521), 122–133.

Tan, C., F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu (2018). A survey on deep transfer learning. In V. Kårková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis (Eds.), *International Conference on Artificial Neural Networks*, pp. 270–279.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Vincent, G., C. Arthur, L. Sylvain, and D. Marcin (2022). A fair pricing model via adversarial learning. *arXiv preprint arXiv:2202.12008*.

Werner, G. and C. Modlin (2016). *Basic Ratemaking*. Cambridge: MIT Press.

Yang, L. and P. Shi (2019). Multiperil rate making for property insurance using longitudinal data. *Journal of the Royal Statistical Society: Series A (Statistics in Society) 182*(2), 647–668.