ORIGINAL RESEARCH PAPER

On clustering levels of a hierarchical categorical risk factor

Bavo D.C. Campo[1] and Katrien Antonio[1,2,3,4]

[1]Faculty of Economics and Business, KU Leuven, Belgium; [2]Faculty of Economics and Business, University of Amsterdam, Amsterdam, The Netherlands; [3]LRisk, Leuven Research Center on Insurance and Financial Risk Analysis, KU Leuven, Belgium; and [4]LStat, Leuven Statistics Research Center, KU Leuven, Belgium
Corresponding author: Bavo D.C. Campo; Email: bavo.campo@kuleuven.be

Abstract

Handling nominal covariates with a large number of categories is challenging for both statistical and machine learning techniques. This problem is further exacerbated when the nominal variable has a hierarchical structure. We commonly rely on methods such as the random effects approach to incorporate these covariates in a predictive model. Nonetheless, in certain situations, even the random effects approach may encounter estimation problems. We propose the data-driven Partitioning Hierarchical Risk-factors Adaptive Top-down algorithm to reduce the hierarchically structured risk factor to its essence, by grouping similar categories at each level of the hierarchy. We work top-down and engineer several features to characterize the profile of the categories at a specific level in the hierarchy. In our workers' compensation case study, we characterize the risk profile of an industry via its observed damage rates and claim frequencies. In addition, we use embeddings to encode the textual description of the economic activity of the insured company. These features are then used as input in a clustering algorithm to group similar categories. Our method substantially reduces the number of categories and results in a grouping that is generalizable to out-of-sample data. Moreover, we obtain a better differentiation between high-risk and low-risk companies.

1. Introduction

At the heart of a risk-based insurance pricing model is a set of risk factors that are predictive of the loss cost. To model the relation between the risk factors and the loss cost, actuaries rely on statistical and machine learning techniques. Both approaches are able to handle different types of risk factors (i.e. nominal, ordinal, geographical, or continuous). In this contribution, we focus on challenges imposed by nominal variables with a hierarchical structure. Such variables may cause estimation problems, due to an exceedingly large number of categories and a limited number of observations for some of the categories. Using default methods to handle these, such as dummy encoding, may result in unreliable parameter estimates in generalized linear models (GLMs) and may cause machine learning methods to become computationally intractable. We refer to this type of risk factor as a hierarchical multi-level factor (MLF) (Ohlsson and Johansson, 2010) or a hierarchical high-cardinality attribute (Micci-Barreca, 2001; Pargent et al., 2022). Examples of such a nominal variable include provinces and municipalities within provinces, or vehicle brands and models within brands. Within workers' compensation insurance, a typical example is the

hierarchical MLF derived from the numerical codes of the Nomenclature générale des Activités économiques dans les Communautés Européennes (NACE) system. The NACE system is a hierarchical classification system used in the European Union to group similar companies based on their economic activity (Eurostat, 2008). A similar example is the Australian and New Zealand Standard Industrial Classification (ANZSIC) system (Australian Bureau of Statistics and New Zealand, 2006), which is closely related to the NACE system (Eurostat, 2008).

In predictive modeling, such risk factors are potentially a great source of information. In workers' compensation insurance, certain industries (e.g., manufacturing, construction) and occupations (e.g., laboring, roofing) are associated with an increased risk of filing claims (Walters et al., 2010; Holizki et al., 2008; Wurzelbacher et al., 2021). Furthermore, companies operating in the same industry are exposed to similar risks. This creates a dependency among companies active in the same industry and heterogeneity between companies working in different industries (Campo and Antonio, 2023). Industry classification systems, such as the NACE and ANZSIC, allow to group companies based on their economic activity at varying levels of granularity. Most industry classification systems are hierarchical classifications that work top-down. The classification of a company starts at the highest level in the hierarchy and, from here, proceeds successively to lower levels in the hierarchy. At the top level of the hierarchy, the categories are broad and general, covering a wide range of economic activities. As we move down the hierarchy, the categories are broken down into increasingly specific subcategories that encompass more detailed economic activities. Further, industry classification systems typically provide a textual description for the categories at all levels in the hierarchy. This description explains why companies are grouped in the same category and can be used to judge the similarity of activities among categories.

To incorporate the hierarchical MLF in a predictive model, we can opt for the hierarchical random effects approach (Campo and Antonio, 2023). Here, we specify a random effect at each level in the hierarchy. The random effects capture the unobservable characteristics of the categories at the different levels in the hierarchy. Moreover, random effects models account for the within-category dependency and between-category heterogeneity. To estimate a random effects model, we can either use the hierarchical credibility model (Jewell, 1975), Ohlsson's combination of the hierarchical credibility model with a GLM (Ohlsson, 2008) or the mixed models framework (Molenberghs and Verbeke, 2005). These estimation procedures rely on the estimation of variance parameters and we require these estimates to be non-negative (Molenberghs and Verbeke, 2011; Oliveira et al., 2017). In some cases, however, we obtain negative variance estimates and this can occur when there is low variability (Oliveira et al., 2017) or when the hierarchical structure of the MLF is misspecified (Pryseley et al., 2011). In these situations, the estimation procedure yields nonsensical results. With the random effects approach, we implicitly assume that the risk profiles differ between the different categories (Tutz and Oelker, 2017). However, it is not an unreasonable assumption that certain categories have an identical effect on the response and that these should be grouped into homogeneous clusters. Decreasing the total number of categories leads to sparser models that are easier to interpret and less likely to experience estimation problems or to overfit. Additionally, individual categories will have more observations, leading to more precise estimates of their effect on the response.

To group data into homogeneous clusters, we typically rely on clustering techniques. These techniques partition the data points into clusters such that observations within the same cluster are more similar compared to observations belonging to other clusters (Hastie et al., 2009, Chapter 14). Within actuarial sciences, clustering methods recently appeared in a variety of applications. In motor insurance, for example, clustering algorithms are employed to group driving styles of policyholders (Wüthrich, 2017; Zhu and Wüthrich, 2021), to construct tariff classes in an unsupervised way (Yeo et al., 2001; Wang and Keogh, 2008) and to bin continuous or spatial risk factors (Henckaerts et al., 2018). Also in health insurance, we find several examples. Rosenberg and Zhong (2022) used clustering techniques to identify high-cost health care utilizers who are responsible for a substantial amount of health expenditures.

Our aim is to use clustering algorithms in workers' compensation insurance pricing, to group categories of the hierarchical MLF that are similar in riskiness and economic activity. To characterize the riskiness, we rely on risk statistics such as the average damage rate and the expected claim frequency. Moreover, we also use the textual description of the categories to obtain information on the economic activity. The risk statistics are expressed as numerical values, whereas the textual descriptions are presented as nominal categories. Both types of features are then used as input in a clustering algorithm to group similar categories of the hierarchical MLF. To create clusters, most algorithms rely on distance or (dis)similarity metrics to quantify the degree of relatedness between observations in the feature space (Hastie et al., 2009; Foss et al., 2019). These metrics, however, are different for numeric and nominal features which makes it challenging to cluster mixed-type data (Cheung and Jia, 2013; Foss et al., 2019; Ahmad et al., 2019). One approach to tackle this problem is to convert the nominal to numeric features. Hereto, we commonly employ dummy encoding (Hsu, 2006; Cheung and Jia, 2013; Ahmad et al., 2019; Foss et al., 2019). This encoding creates binary variables that represent category membership. Hereby, it results in a loss of information when the categories have textual labels. Labels provide meaning to categories and reflect the degree of similarity between different categories. A more suitable encoding is obtained using embeddings, a technique developed within natural language processing (NLP). Embeddings are vector representations of textual data that capture the semantic information (Verma et al., 2021; Schomacker and Tropmann-Frick, 2021; Ferrario and Naegelin, 2020). In the actuarial literature, Lee et al. (2020) show how embeddings can be used to incorporate textual data into insurance claims modeling. Xu et al. (2022) create embedding-based risk factors that are used as features in a claim severity model. Zappa et al. (2021) used embeddings to create risk factors that predict the severity of injuries in road accidents.

Research on grouping categories of hierarchical MLFs is limited. Most of the research is focused on nominal variables that do not have a hierarchical structure. For example, the Generalized Fused Lasso (GFL) (Höfling et al., 2010; Gertheiss and Tutz, 2010; Oelker et al., 2014) groups MLF categories within a regularized regression framework. Here, categories are merged when there is a small difference between the regression coefficients. Nonetheless, the GFL does not scale well to high-cardinality features since the number of estimated coefficient differences grows exponentially with the number of categories. Another example to fuse non-hierarchically structured nominal variables is the method of Carrizosa et al. (2021). Here, the authors first specify the order of the categories and the number of clusters. Next, to create the prespecified number of clusters they group consecutive categories. For a specific number of clusters, multiple solutions exist and the solution with the highest out-of-sample accuracy is preferred. The disadvantages of this approach are three-fold. First, it only merges neighboring categories. Second, there is no procedure to select the optimal number of groups and third, the procedure can not immediately be applied to hierarchical MLFs. To the best of our knowledge, the method described by Carrizosa et al. (2022) is the only approach that focuses on reducing the number of categories of a hierarchical MLF. Here, the authors propose a bottom-up clustering strategy. This technique begins by considering the categories at the lowest level in the hierarchy. Hereafter, starting from the categories at the lowest level in the hierarchy, they consecutively merge categories at the lowest level into broader categories at higher levels in the hierarchy. As such, categories that are nested within the same category at a higher level in the hierarchy are grouped. Notwithstanding, their proposed optimization strategy is only suited for linear regression models. Further, it is not suitable when we want to maintain the levels in the hierarchical structure. Certain granular categories are replaced by the broader categories they are nested in. Hence, it is possible that the optimal solution produces a hierarchical structure with a different depth in its representation.

Our paper contributes to the existing literature in the following ways. Firstly, we present a data-driven approach to reduce an existing granular hierarchical structure to its essence, by grouping similar categories at every level in the hierarchy. We devise a top-down procedure, where we start

at the top level, to preserve the hierarchical structure. At a specific level in the hierarchy, we engineer several features to characterize the risk profile of each category. In a case study with a workers' compensation insurance product, we use predicted random effects obtained with a generalized linear mixed model for damage rates on the one hand and claim frequencies on the other hand. To extract the textual information contained in the category description, we use embeddings. Next, we use these features as input in a clustering algorithm to group similar categories into clusters. Hereafter, we proceed to group the categories at the next hierarchical level. The procedure stops once we have grouped the categories at the lowest level in the hierarchy. Secondly, we provide a concise overview of important aspects and algorithms in cluster analysis. Furthermore, we demonstrate that the clustering algorithm and evaluation criterion affect the clustering solution. Thirdly, we show that embeddings can be used to group similar categories of a nominal variable. Contrary to Lee et al. (2020); Zappa et al. (2021); and Xu et al. (2022), we do not employ embeddings to create new risk factors in a pricing model. Instead, we use embeddings to extract the textual information of category labels and to cluster categories based on their semantic similarities.

The remainder of this paper is structured as follows. In Section 2, we use a workers' compensation insurance product as a motivating example with the NACE code as hierarchical MLF. We illustrate the structure of this type of data set, which information is typically available and we explain how to engineer features that characterize the risk profile of the categories. In Section 3, we define a top-down procedure to cluster similar categories at a specific level in the hierarchy and we discuss several aspects of clustering techniques. The results of applying our procedure to reduce the hierarchical structure to its essence are discussed in Section 4. Moreover, in this section, we also compare the use of the original and the reduced structure in a technical pricing model for workers' compensation insurance. Section 5 concludes the article.

2. Feature engineering for industrial activities in a workers' compensation insurance product

To illustrate the importance of hierarchical MLFs in insurance pricing, a workers' compensation insurance product is a particularly suitable example. This insurance product compensates employees for lost wages and medical expenses resulting from job-related injury (see Campo and Antonio (2023) for more information). In this type of insurance, we generally work with an industrial classification system to group companies based on their economic activity. Hereto, we demonstrate and discuss the NACE classification system in Section 2.1. We explain the typical structure of a workers' compensation insurance data set and discuss which information is available. In Section 2.2, we show how we use this information to engineer features that characterize the riskiness and economic activity of categories at a specific level in the hierarchy.

2.1 A hierarchical classification scheme for industrial activities

In a workers' compensation insurance data set, it is common to work with an industrial classification system. Within the European Union, a wide range of organizations (e.g., national statistical institutes, business and trade associations, insurance companies, and European national central banks (Eurostat, 2008; Stassen et al., 2017; European Central Bank, 2021)) work with the NACE system (Eurostat, 2008). This is a hierarchical classification system to group companies based on their economic activity. Each company is assigned a four-digit numerical code, which is used to identify the categories at different levels in the hierarchy. The NACE system works top-down, starting at the highest level in the hierarchy and then proceeding to the lower levels. In our paper, we work with NACE Rev. 1 (Statistical Office of the European Communities, 1996), which has five hierarchical levels (in descending order): `section`, `subsection`, `division`, `group`, and `class`. Most member states of the EU have a national version that follows the same structural and hierarchical framework as the NACE. In Belgium, the national version of NACE Rev. 1 is called the
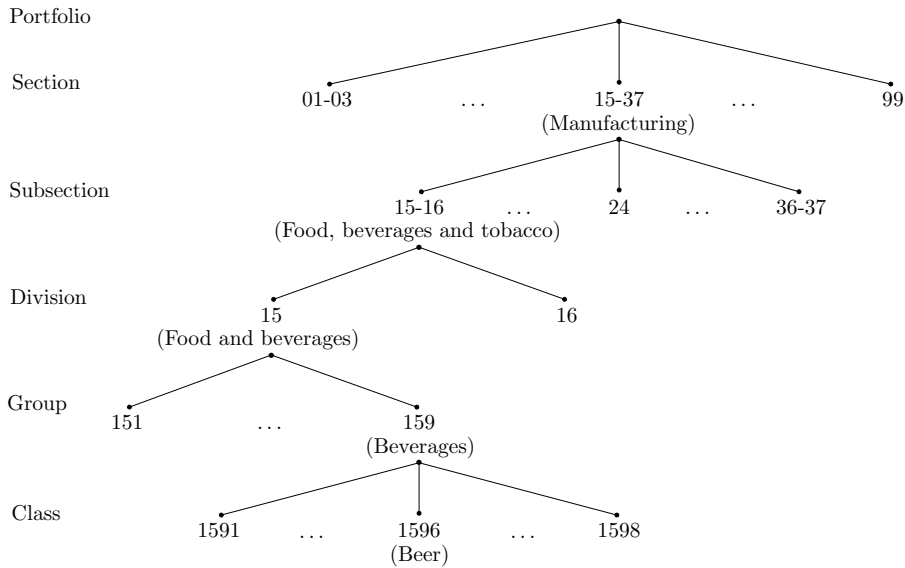
Figure 1. Illustration of the NACE system for a company that manufactures beer. This economic activity is encoded as 1596 in NACE Rev. 1. Based on the NACE code, we assign the company to a certain category at a specific level in the hierarchy. For the purpose of this illustration, we shortened the textual description of the categories.

NACE-Bel (2003) (FOD Economie, 2004) and adds one more level to the hierarchy by adding a fifth digit. We refer to this level as `subclass`. Insurance companies may choose to add a sixth digit to include yet another level, allowing for even more differentiation between companies. This level in the hierarchy is referred to as `tariff group` and we denote the insurance company's version as NACE-Ins.

To illustrate how the NACE system works, suppose that we have a company that manufactures beer. Using NACE Rev. 1 (Statistical Office of the European Communities, 1996), this company gets the code 1596. At the top level `section`, the first two digits (i.e. 15) classify this company into *manufacturing* (see Fig. 1). This category contains all NACE codes that start with numbers 15 to 37. Following, the first two digits categorize the company into *manufacture of food products, beverages, and tobacco* at the `subsection` level, which is nested within the `section` *manufacturing*. The category *manufacture of food products, beverages, and tobacco* includes all NACE codes starting with numbers 15 and 16. At the third level in the hierarchy – `division` – the company is classified into *15 – manufacture of food products and beverages*. At the fourth level `group`, we use the first three digits to classify the company in *159 - manufacture of beverages*. Finally, at the fifth and lowest level `class`, the four-digit code assigns the company to *1596 – manufacture of beer*.

This example also shows that, at all levels in the hierarchy, the NACE provides a textual description for each category. This text briefly describes the economic activity of a specific category, thereby explaining why certain companies are grouped. We illustrate this in Table 1. Herein, we show the textual information that is available for companies with NACE codes 1591, 1596, and 1598. This table displays a separate column for every level in the hierarchy. The corresponding value indicates which category the codes belong to. At the `section`, `subsection`, `division`, and `group` level, the NACE codes 1591, 1596, and 1598 are grouped in the same categories (i.e. `15-37`, `15-16`, `15`, and `159`, respectively). At the `class` level, each of the codes is assigned to a different category. The column description presents the textual information for each corresponding category. For example, at the `division` level, 15 has the description *manufacture of food products and beverages*.

Table 1. Illustration of the textual information for NACE codes 1591, 1596, and 1598

Section	Subsection	Division	Group	Class	Description
15-37					Manufacturing
	15-16				Manufacture of food products, beverages and tobacco
		15			Manufacture of food products and beverages
			159		Manufacture of beverages
				1591	Manufacture of distilled potable alcoholic beverages
				. . .	. . .
				1596	Manufacture of beer
				. . .	. . .
				1598	Production of mineral waters and soft drinks

Table 2. Number of unique categories per level in the hierarchy of the NACE-Bel (2003)

	Section	Subsection	Division	Group	Class	Subclass
Number of categories:						
NACE-Bel (2003)	17	31	62	224	515	800
Portfolio	17	30	56	197	398	581

The textual description of other categories in the NACE system is similar to the example in Table 1. Overall, the description is brief and consists of a single word or phrase. Further, categories at higher levels in the hierarchy have a more concise description using overarching terms such as *manufacturing*. Conversely, at lower levels in the hierarchy, the descriptions are typically more detailed and extensive (e.g., *manufacture of distilled potable alcoholic beverages* and *production of mineral waters and soft drinks*).

2.1.1 Selecting levels in the hierarchy

When working with NACE-Ins, we have seven levels in the hierarchy: `section`, `subsection`, `division`, `group`, `class`, `subclass`, and `tariff group`. This results in an immense amount of categories, some of which contain few to no observations. In this paper, we work with a NACE-Ins that is based on the NACE-Bel (2003) (FOD Economie, 2004). Table 2 shows the unique number of categories at each level in the hierarchy. The first row indicates how many categories there are at each level in the NACE-Bel (2003) classification system. The second row specifies how many categories are present at each level in our portfolio. Due to the confidentiality of the data, we do not disclose the number of categories at the `tariff group` level.

The more levels in the hierarchy, the more complex a tariff model becomes that incorporates this hierarchical MLF in its full granularity. Consequently, in practice, the NACE-Ins may not be used in its entirety. In our database, we have access to a hierarchical MLF designed by the insurance company that offers the workers' compensation insurance product. This structure has been created by merging similar NACE-Ins codes, based on expert judgment. This hierarchical MLF has two levels in the hierarchy, referred to as `industry` and `branch` in Campo and Antonio (2023). In our paper, we illustrate how such a hierarchical MLF can be constructed using our proposed data-driven approach, as an alternative for the manual grouping by experts. To demonstrate our method we put focus on two selected levels in the hierarchy of NACE-Ins. This is merely for illustration purposes. Our approach is applicable with any number of levels in the hierarchy.

To align with the insurance company's hierarchical MLF, we select the `subsection` and `tariff group` level (see Fig. 1) and aim to reduce the hierarchical structure consisting of these

levels. We use $l = (1, \dots, L)$ to index the levels in the hierarchy, where L denotes the total number of levels. In our illustration, `subsection` corresponds to the highest level $l = 1$ in the hierarchy. At $l = 1$, we index the categories using $j = (1, \dots, J)$ where J denotes the total number of categories. The `tariff group` level represents the second level in the hierarchy. Here, we use $jk = (j1, \dots, jK_j)$ to index the categories nested within `subsection` j. We refer to k as the child category that is nested within parent category j and K_j denotes the total number of categories nested within j. Due to confidentiality of the classification system and data, no comparisons between the company's hierarchical MLF and our proposed clustering solutions will be provided in the paper.

2.2 Feature engineering

We start at the highest level in the hierarchy, `subsection`, and engineer a set of features that capture the riskiness and the economic activity of the categories. Using these features, the clustering algorithms can identify and group similar categories at the `subsection` level. Hereafter, we proceed to the `tariff group` level and engineer the same type of features for the categories at this level in the hierarchy.

We assume that we have a workers' compensation insurance data set with historical, claim-related information of the companies in our portfolio. For each company i, we have the NACE-Ins code in year t. We use this code to categorize company i in `subsection` j and `tariff group` k. In our database, we have the total claim amount Z_{ijkt}, the number of claims N_{ijkt}, and the salary mass w_{ijkt} for year t and company i, a member of `subsection` j and `tariff group` k.

2.2.1 Riskiness

We express the riskiness of the `subsection` and `tariff group` in terms of their damage rate and their claim frequency. The higher the damage rate and claim frequency, the riskier the category. For an individual company i, the damage rate in year t is calculated as

$$Y_{ijkt} = \frac{Z_{ijkt}}{w_{ijkt}}. \tag{1}$$

To capture the `subsection`- and `tariff group`-specific effect on the damage rate and claim frequency, we use random effects models (Campo and Antonio, 2023). In this approach, the prediction of the category-specific random effect is dependent on how much information is available. The random effect predictions are shrunk toward zero for categories with high variability, a low number of observations, or when the variability between categories is small (Breslow and Clayton, 1993; Gelman and Hill, 2017; Brown and Prescott, 2006; Pinheiro et al., 2009).

We start at the `subsection` level and model the damage rate as a function of the `subsection` using a (Tweedie generalized) linear mixed model

$$g\left(E\left[Y_{ijkt} | U_j^d \right] \right) = \mu^d + U_j^d. \tag{2}$$

Here, μ^d denotes the intercept and U_j^d the random effect of `subsection` j. We include the salary mass w_{ijkt} as weight. As discussed in Campo and Antonio (2023), we typically model the damage rate by assuming either a Gaussian or Tweedie distribution for the response. The U_j^d's represent the between-`subsection` variability and enable us to discern between low- and high-risk profiles. The higher U_j^d, the higher the expected damage rate for `subsection` j and vice versa. To engineer the first feature, we extract the $\widehat{U}_j^d$'s from the fitted damage rate model (2). Hereafter, we fit a Poisson generalized linear mixed model

$$g\left(E\left[N_{ijkt} | U_j^f \right] \right) = \mu^f + U_j^f + \log(w_{ijkt}) \tag{3}$$

to assess the `subsection`'s effect on the claim frequency. μ^f denotes the intercept and U_j^f the random effect of `subsection` j. We include the log of the salary mass as an offset variable. We extract the $\widehat{U}_j^f$'s from the fitted claim frequency model (3) to engineer the second feature. The $\widehat{U}_j^d$'s and $\widehat{U}_j^f$'s will be combined with the features representing the economic activity to group similar categories at the `subsection` level. We index the resulting grouped categories at the `subsection` level using $j' = (1, \ldots, J')$.

Next, we engineer the features at the `tariff group` level. To capture the `tariff group`-specific effect on the damage rate, we fit a (Tweedie generalized) linear mixed model

$$g\Big(E\Big[Y_{ij'kt}|U_{j'}^d, U_{j'k}^d\Big]\Big) = \mu^d + U_{j'}^d + U_{j'k}^d \tag{4}$$

where the random effect $U_{j'k}^d$ represents the `tariff group`-specific deviation from $\mu^d + U_{j'}^d$. As before, we include the $w_{ij'kt}$ as weight. The $U_{j'k}^d$'s reflect the between-`tariff group` variability, after having accounted for the variability between the grouped `subsections`. $U_{j'k}^d$ quantifies the `tariff group`-specific effect of `tariff group` k, in addition to the (grouped) `subsection`-specific effect $U_{j'}^d$, on the expected damage rate. To characterize the riskiness of the categories at the `tariff group` level in terms of the claim frequency, we extend (3) to

$$g\Big(E\Big[N_{ij'kt}|U_{j'}^f, U_{j'k}^f\Big]\Big) = \mu^f + U_{j'}^f + U_{j'k}^f + \log(w_{ij'kt}). \tag{5}$$

In this model, $U_{j'k}^f$ represents the `tariff group`-specific deviation from $\mu^d + U_{j'}^f$. We extract the $\widehat{U}_{j'k}^d$'s and $\widehat{U}_{j'k}^f$'s from the fitted damage rate (4) and claim frequency model (5) to engineer the features at the `tariff group` level.

We do not include any additional covariates in the random effects models (see equations (2) to (5) to fully capture the variability that is due to heterogeneity between categories. If, however, there are covariates available at the `subsection` or `tariff group` level, these can be incorporated in the model specification in equations (2) to (5) and used further in the construction of the feature matrix at the `subsection` or `tariff group` level.

The approach to construct features based on the random effect predictions is closely related to target encoding (Micci-Barreca, 2001). In target encoding, the numerical value of a category is the weighted average of the category-specific average of the response variable and the response variable's average at the higher levels in the hierarchy. Micci-Barreca (2001) presents various approaches to determine the weights. A linear mixed model can be seen as a special case of the weights specification as it has a closed-form solution for the random effects predictions. For most GLMMs, however, there is no available analytical expression. In these cases, there is no strict equivalence with target encoding.

2.2.2 Economic activity

Next, we require a feature that expresses similarity in economic activity. Given that industry codes of similar activities tend to be closer, we might rely on their numerical values. The closer the numerical value, the more overlap there will be between categories at a specific level in the hierarchy. Hence, we could use the four-digit NACE codes as discussed in Section 2.1. Notwithstanding, not all categories can readily be converted to a numerical format. At the higher levels in the hierarchy, we have categories that encompass various NACE codes, such as *manufacture of pulp, paper, and paper products: publishing and printing* at the `subsection` level. This specific `subsection` consists of all NACE codes that start with numbers 21 to 22. To obtain a numerical representation of this `subsection` we can, for example, take the mean of these NACE codes. We then obtain the

Table 3. Illustration of a possible encoding for the categories at the `subsection` level of the NACE Rev. 1

Subsection	Encoding	Description
...	...	...
20	20	Manufacture of wood and wood products
21-22	21.5	Manufacture of pulp, paper and paper products: publishing and printing
23	23	Manufacture of coke, refined petroleum products and nuclear fuel
24	24	Manufacture of chemicals, chemical products and mad-made fibers
...	...	...
36-37	36.5	Manufacturing not elsewhere classified
40-41	40.5	Electricity, gas and water supply
45	45	Construction
...	...	...

encodings as illustrated in Table 3. The column `subsection` indicates which category the NACE codes are appointed to at the `subsection` level and the column description gives the textual description of this category. The examples in this table highlight two issues with this approach. Firstly, the numerical distance may not reflect the similarity between economic activities. The difference between *manufacture of wood and wood products* and *manufacture of pulp, paper, and paper products: publishing and printing* is the same as the difference between *manufacture of pulp, paper, and paper products: publishing and printing* and *manufacture of coke, refined petroleum products, and nuclear fuel*. Hence, using this encoding would imply that *manufacture of pulp, paper, and paper products: publishing and printing* is as similar to *manufacture of wood and wood products* as it is to *manufacture of coke, refined petroleum products, and nuclear fuel*. Secondly, there might be gaps between consecutive codes in the classification system. In NACE Rev. 1, for example, there are no codes that start with 43 or 44. Gaps such as these can be present at various levels in the hierarchy and generally exist to allow for future additional categories (Eurostat, 2008).

Alternatively, we use embeddings to encode the economic activity of a category (Mikolov et al., 2013). Embeddings map the textual information to a continuous vector. Hence, we use embeddings to map the description for `subsection` j to a vector $e_j = (e_{j1}, e_{j2}, \ldots, e_{jE})$. For `tariff group` k within `subsection` j, we denote the embedding vector as $e_{jk} = (e_{jk1}, e_{jk2}, \ldots, e_{jkE})$. Here, E is the dimension of the vector which depends on the encoder. The textual information from similar categories is expected to lie closer in the vector space and this enables us to group semantically similar texts (i.e. texts that are similar in meaning).

Consequently, we group categories with comparable economic activities based on the embeddings of their textual labels. To engineer these embeddings, we may train an encoder on a large corpus of text. Within the area of NLP, researchers commonly use neural networks (NN) as encoders. By training the NN on large amounts of unstructured text data, it is able to learn high-quality vector representations (Mikolov et al., 2013). In addition, the encoders can be trained to learn vector representations for words, phrases, or paragraphs. The disadvantage of these models is that they generally need a large amount of data (Arora et al., 2020; Troxler and Schelldorfer, 2022). Furthermore, words that do not appear often are poorly represented (Luong et al., 2013). We, therefore, prefer pre-trained encoders, which are trained on large text corpuses, to encode the textual description of a `subsection`. For example, the universal sentence encoder of Cer et al. (2018) is trained on Wikipedia, web news, web question-answer pages, and discussion forums. This encoder is publicly available via TensorFlow Hub (https://tfhub.dev/).

Table 4. Feature matrix $\mathcal{F}_1$, consisting of the engineered features for the categories at $l = 1$ in the hierarchy. The columns $_1\widehat{\boldsymbol{U}}^d$ and $_1\widehat{\boldsymbol{U}}^f$ contain the predicted random effects of the damage rate and claim frequency GLMM, respectively. The embedding vector is represented by the values in columns $\boldsymbol{e}_{\star 1}, \boldsymbol{e}_{\star 2}, \ldots, \boldsymbol{e}_{\star E}$.

Subsection	$_1\widehat{\boldsymbol{U}}^d$	$_1\widehat{\boldsymbol{U}}^f$	$\boldsymbol{e}_{\star 1}$	$\boldsymbol{e}_{\star 2}$	$\boldsymbol{e}_{\star 3}$	$\ldots$	$\boldsymbol{e}_{\star E}$
1	-1.25	-0.25	-2.13	1.25	0.15	$\ldots$	-0.05
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
J	0.75	0.15	1.79	-2.13	0.5	$\ldots$	1.07

Table 5. Feature matrix $\mathcal{F}_2$, consisting of the engineered features for the categories at $l = 2$ in the hierarchy. The columns $_2\widehat{\boldsymbol{U}}^d$ and $_2\widehat{\boldsymbol{U}}^f$ contain the predicted random effects of the damage rate and claim frequency GLMM, respectively. The embedding vector is represented by the values in columns $\boldsymbol{e}_{\star\star 1}, \boldsymbol{e}_{\star\star 2}, \ldots, \boldsymbol{e}_{\star\star E}$.

Subsection	Tariff group	$_2\widehat{\boldsymbol{U}}^d$	$_2\widehat{\boldsymbol{U}}^f$	$\boldsymbol{e}_{\star\star 1}$	$\boldsymbol{e}_{\star\star 2}$	$\boldsymbol{e}_{\star\star 3}$	$\ldots$	$\boldsymbol{e}_{\star\star E}$
1	11	-1.55	-0.01	-0.54	1.08	2.12	$\ldots$	0.10
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
1	$1K_1$	0.15	0.96	-0.37	-0.26	0.58	$\ldots$	-0.99
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
j	$j1$	-0.29	-0.41	-0.05	-0.72	0.41	$\ldots$	-0.73
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
j	jK_j	0.11	0.26	0.16	0.69	0.87	$\ldots$	1.59
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
J	$J1$	0.11	0.26	0.16	0.69	0.87	$\ldots$	1.59
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
J	JK_J	0.67	-1.74	0.19	0.45	0.5	$\ldots$	-0.61

2.2.3 Feature matrix

After engineering the features at level l in the hierarchy, we assemble them into a feature matrix $\mathcal{F}_l$. Table 4 shows an example of the feature matrix $\mathcal{F}_1$ at the `subsection` level. Herein, $_1\widehat{\boldsymbol{U}}^d = (\widehat{U}_1^d, \ldots, \widehat{U}_j^d, \ldots, \widehat{U}_J^d)$ represents the vector with the predicted random effects from the fitted damage rate GLMM (see (2)). $_1\widehat{\boldsymbol{U}}^f = (\widehat{U}_1^f, \ldots, \widehat{U}_j^f, \ldots, \widehat{U}_J^f)$ denotes the vector with the predicted random effects of the claim frequency GLMM (see (3)) and we use the notation $\boldsymbol{e}_{\star 1} = (e_{11}, \ldots, e_{j1}, \ldots, e_{J1})$ for the embeddings. Each row in $\mathcal{F}_1$ corresponds to a numerical representation of a specific `subsection` j, with $j = (1, \ldots, J)$. We denote the feature vector of `subsection` j by $\boldsymbol{x}_j = (\widehat{U}_j^d, \widehat{U}_j^f, \boldsymbol{e}_j)$.

At the `tariff group` level, we gather the features in $\mathcal{F}_2$. An example hereof is given in Table 5. $_2\widehat{\boldsymbol{U}}^d = (\widehat{U}_{11}^d, \ldots, \widehat{U}_{jk}^d, \ldots, \widehat{U}_{JK_J}^d)$ and $_2\widehat{\boldsymbol{U}}^f = (\widehat{U}_{11}^f, \ldots, \widehat{U}_{jk}^f, \ldots, \widehat{U}_{JK_J}^f)$ denote the vectors of the `tariff group`-specific random effects. To denote the embeddings, we use $\boldsymbol{e}_{\star\star 1} = (e_{111}, \ldots, e_{jk1}, \ldots, e_{JK_J1})$. $\boldsymbol{x}_{jk} = (\widehat{U}_{jk}^d, \widehat{U}_{jk}^f, \boldsymbol{e}_{jk})$ denotes the feature vector of `tariff group` k, nested within `subsection` j.

3. Clustering levels in a hierarchical categorical risk factor

3.1 Partitioning hierarchical risk-factors adaptive top-down

To group similar categories at each level in the hierarchy, we devise the PHiRAT algorithm to **P**artition **Hi**erarchical **R**isk-factors in an **A**daptive **T**op-down way, see Algorithm 1. We introduce

Algorithm 1. PHiRAT Pseudo-code

---

for $l = 1$ *to* L **do**

    Engineer features that characterize the categories;

    Combine the features in a feature matrix $\mathcal{F}_l$;

    if $l = 1$ **then**

        Use a clustering algorithm to group the $(1, \ldots, J)$ categories into $\mathcal{K}_1$ clusters, with $\mathcal{F}_1$ as input;

    else

        foreach p *in* $\mathcal{J}_{l-1}$ **do**

            Use a clustering algorithm to group the $\{c : \pi(c) = p\}$ child categories of parent category p into $\mathcal{K}_{l,\{c:\pi(c)=p\}}$ clusters, using $\mathcal{F}_{l,\{c:\pi(c)=p\}}$ as input;

        end

end

---

some additional notation to explain how PHiRAT works. $\mathcal{J}_l$ denotes the set of categories at a specific level l in the hierarchy. Hence, when the total number of levels $L = 2$, $\mathcal{J}_1 = (1, \ldots, J)$ and $\mathcal{J}_2 = (11, \ldots, 1K_1, \ldots, j1, \ldots, jK_j, \ldots, J1, \ldots, JK_J)$ as illustrated in Tables 4 and 5, respectively. We use $\pi(c) = p$ to indicate that child category c has parent category p and $\{c : \pi(c) = p\}$ denotes the set of all child categories c nested within parent category p. $\mathcal{F}_{l,\{c:\pi(c)=p\}}$ represents the subset of feature matrix $\mathcal{F}_l$, which contains only those rows of $\mathcal{F}_l$ that correspond to the child categories nested in parent category p. For example, when $L = 2$, $\mathcal{F}_{2,\{c:\pi(c)=j\}}$ contains only the rows corresponding to the $(j1, \ldots, jK_j)$ child categories of j (see Table 5). Further, $\mathcal{K}_l$ denotes the number of clusters at level l in the hierarchy. For $l > 1$, we extend the notation to $\mathcal{K}_{l,\{c:\pi(c)=p\}}$ to indicate that, at level l in the hierarchy, we group the child categories of p into $\mathcal{K}_{l,\{c:\pi(c)=p\}}$ clusters. The subscript l indicates at which level in the hierarchy we are. We use the additional subscript $\{c : \pi(c) = p\}$ to specify that we only consider the set of child categories of parent category p.

PHiRAT works top-down, starting from the highest level ($l = 1$) and working its way down to the lowest level ($l = L$) in the hierarchy (see Algorithm 1). Every iteration consists of three steps. First, we engineer features for the categories in $\mathcal{J}_l$. Second, we combine these features in a feature matrix $\mathcal{F}_l$. Third, we employ clustering techniques to group the categories in $\mathcal{J}_l$ using (a subset of) $\mathcal{F}_l$ as input. In most clustering methods, we define a tuning grid and perform a grid search to determine the optimal number of clusters. Consequently, the minimum value within the tuning grid sets the lower bound for the number of grouped categories. Further, the third step differs slightly when $l = 1$. At $l = 1$, we use the full feature matrix as input in the clustering algorithm. Conversely, when $l > 1$, we loop over the parent categories in $\mathcal{J}_{l-1}$ and in every loop, we use a different subset of $\mathcal{F}_l$ as input. For parent category p, we only consider $\{c : \pi(c) = p\}$ and use $\mathcal{F}_{l,\{c:\pi(c)=p\}}$ as input in the clustering algorithm. Hereby, we ensure that we only group child categories nested within parent category p. Further, we remove a specific level l from the hierarchy if $\mathcal{J}_l$ is reduced to $\mathcal{J}_{l-1}$ (i.e. all child categories, of each parent category in $\mathcal{J}_{l-1}$, are merged into a single group). The algorithm stops when the clustering at level $l = L$ is done.

We visualize how the procedure works in Fig. 2. In this fictive example, we focus on the `subsection` and `tariff group` level. At the `subsection` level, there are three unique categories $\mathcal{J}_1 = (45, 80, 85)$ and at the `tariff group` level, we have nine unique categories $\mathcal{J}_2 =$ $(451100, 453100, 454500, 801000, 802100, 803000, 851400, 852000, 853100)$ (see Fig. 2a). We use the index $j = (1, \ldots, J)$ at $l = 1$. At $l = 2$, we use $jk = (j1, \ldots, jK_j)$ to index the categories nested within j. Using PHiRAT, we first group the categories 80 and 85 at $l = 1$. This is depicted in Fig. 2a by the blue rectangle. Consequently, $\mathcal{J}_1 = (45, \{80;85\})$ where $\{80; 85\}$ denotes that categories 80 and 85 are merged. The grouped categories at $l = 1$ are now indexed by $j' = (1, \ldots, J')$. Hereafter,

(a)

Visualization of the hierarchically structured categories at the `subsection` and `tariff group` level before clustering. The blue rectangles depict which categories are grouped when employing the PHiRAT algorithm.
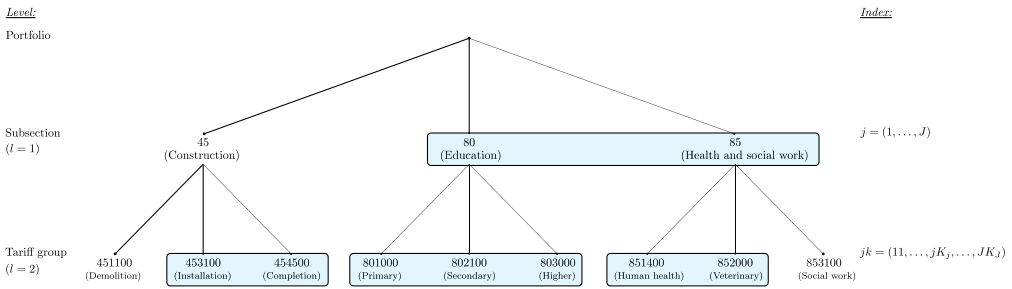
(b)

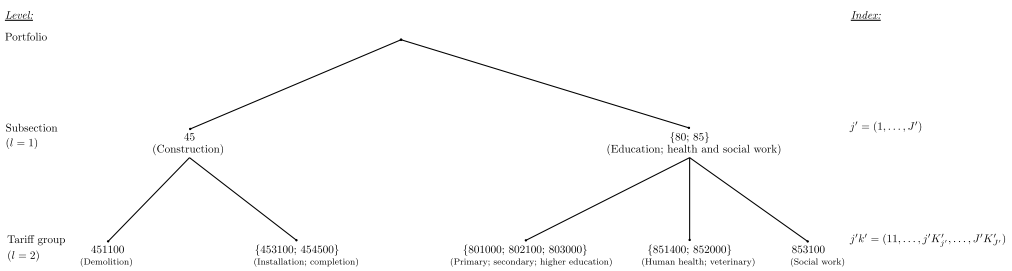Visualization of the reduced hierarchical structure after clustering with the PHiRAT algorithm.

Figure 2. A fictive example illustrating how the PHiRAT algorithm clusters categories at the `subsection` and `tariff group` level. The textual labels of the categories are shortened for the purpose of this illustration.

the algorithm iterates over the (fused) categories in $\mathcal{J}_1$ and clusters the child categories at $l = 2$. Within `subsection` 45, it groups the categories 453100 and 454500. Within {80;85}, the clustering results in three groups of categories: (1) {801000;802100;803000}; (2) {851400;852000} and (3) 853100. We index the fused categories at $l = 2$ and nested within j' using $j'k' = (j'1, \ldots, j'K'_{j'})$. At this point the algorithm stops and Fig. 2b depicts the clustering solution.

We opt for a top-down approach for several reasons. Firstly, at the highest level we have more observations available for the categories. The more data, the more precise the category-specific risk estimates will be. Conversely, categories at more granular levels have fewer observations, leading to less precise risk estimates. Secondly, we preserve the original hierarchical structure and maintain the parent-child relationship between categories at different levels in the hierarchy. Hereby, we divide the grouping of categories, at a specific level in the hierarchy, into smaller and more specific separate clustering problems.

Alternatively, it might be interesting to construct a similar algorithm that works bottom-up and groups child categories that have different parent categories. We leave this as a topic for future research.

3.2 Clustering analysis

To partition the set of categories $\mathcal{J}_l$ into homogeneous groups, we rely on clustering algorithms using (a subset of) $\mathcal{F}_l$ as input. At $l = 1$, for example, we employ clustering to divide the rows in $\mathcal{F}_1$ into $\mathcal{K}_1$ homogeneous groups such that categories in each cluster j' are more similar to each other compared to categories of other clusters $j' \neq j'$. In the remainder of this section, we continue with the example of grouping the $(1, \ldots, J)$ categories at $l = 1$ to explain and illustrate the key concepts of the clustering methods used in this paper.

Table 6. Overview of existing (dis)similarity metrics to quantify the proximity between observations. We select the angular similarity and angular distance, as they are better suited to measure the similarity between embeddings and also compatible with clustering algorithms

	Dissimilarity	Similarity
Euclidean[a]	$\|\boldsymbol{x}_j - \boldsymbol{x}_{\jmath}\|_2^2$	$\exp\left(\dfrac{-\|\boldsymbol{x}_j - \boldsymbol{x}_{\jmath}\|_2^2}{\sigma^2}\right)$
Cosine	$1 - \dfrac{\boldsymbol{x}_j^\top \boldsymbol{x}_{\jmath}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{\jmath}\|_2}$	$\dfrac{\boldsymbol{x}_j^\top \boldsymbol{x}_{\jmath}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{\jmath}\|_2}$
Angular	$\pi^{-1} \cos^{-1}\left(\dfrac{\boldsymbol{x}_j^\top \boldsymbol{x}_{\jmath}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{\jmath}\|_2}\right)$	$1 - \pi^{-1} \cos^{-1}\left(\dfrac{\boldsymbol{x}_j^\top \boldsymbol{x}_{\jmath}}{\|\boldsymbol{x}_j\|_2 \cdot \|\boldsymbol{x}_{\jmath}\|_2}\right)$

[a]σ is a scaling parameter set by the user (Ng et al., 2001; Poon et al., 2012).

Most clustering algorithms rely on distance or (dis)similarity metrics to quantify the proximity between observations. Table 6 gives an overview of a selected set of (dis)similarity measures relevant to our paper. A dissimilarity measure $d(\boldsymbol{x}_j, \boldsymbol{x}_{\jmath})$ expresses how different two observations $\boldsymbol{x}_j$ and $\boldsymbol{x}_{\jmath}$ are (the higher the value, the more they differ). Dissimilarity metrics that satisfy the triangle inequality $d(\boldsymbol{x}_j, \boldsymbol{x}_{\jmath}) \leqslant d(\boldsymbol{x}_j, \boldsymbol{x}_z) + d(\boldsymbol{x}_z, \boldsymbol{x}_{\jmath})$ for any z (Schubert, 2021; Phillips, 2021) are considered proper distance metrics. Most dissimilarity metrics can easily be converted to a similarity measure $s(\cdot, \cdot)$ which expresses how comparable two observations are, with similar observations obtaining higher values for these measures. The most commonly used dissimilarity metric is the squared Euclidean distance $\|\boldsymbol{x}_j - \boldsymbol{x}_{\jmath}\|_2^2$. Here, $\|\boldsymbol{x}_j\|_2 := \sqrt{x_{j1}^2 + \cdots + x_{jn_f}^2}$ and n_f denotes the number of features considered. Euclidean-based (dis)similarity measures, however, are not appropriate to capture the similarities between embeddings (Kogan et al., 2005). Within NLP, the cosine similarity is, therefore, most often used to measure the similarity between embeddings (Mohammad and Hirst, 2012; Schubert, 2021). Notwithstanding, the cosine similarity ranges from -1 to 1, and in cluster analysis, we generally require the (dis)similarity measure to be non-negative (Everitt et al., 2011; Kogan et al., 2005; Hastie et al., 2009). In this case, we can convert the cosine similarity to the angular similarity, which is restricted to [0, 1]. The angular similarity can be converted to the angular distance, which is a proper distance metric. Conversely, the cosine dissimilarity is not a distance measure since it does not satisfy the triangle inequality. In our study, we therefore rely on the angular similarity and angular distance to group comparable categories.

Using a selected (dis)similarity metric, we compute the proximity measure between all pairwise observations in the matrix with input features. We combine all values in a $J \times J$ similarity matrix S or dissimilarity matrix D, which is used as input in a clustering algorithm. Most algorithms require these matrices to be symmetric (Hastie et al., 2009). In literature on unsupervised learning algorithms, clustering methods are typically divided into three different types: combinatorial algorithms, mixture modeling, and mode seeking (Hastie et al., 2009). Both the mixture modeling and mode-seeking algorithms rely on probability density functions. Conversely, the combinatorial algorithms do not rely on an underlying probability model and work directly on the data. In our paper, we opt for a distribution-free approach and therefore focus on the combinatorial algorithms summarized in Table 7. For the interested reader, a detailed overview of these algorithms is given Appendix B.

In most clustering techniques, the number of clusters $\mathcal{K}$ can be considered a tuning parameter that needs to be carefully chosen from a range of possible (integer) values. Hereto, we require a cluster validation index to select the value for $\mathcal{K}$ which results in the most optimal clustering solution. We divide the cluster validation indices into two groups: internal and external (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019; Halkidi et al., 2001). Using external

Table 7. Overview of clustering algorithms, together with their strengths and drawbacks

Algorithm	Strengths	Drawbacks
k-means (MacQueen et al., 1967)	— Well-known — Simple and easy to implement — Computationally efficient	— Only suited for numeric features — Sensitive to outliers and the initialization — Local optima
k-medoids (Kaufman and Rousseeuw, 1990b)	— Applicable to any feature type — Less sensitive to outliers	— Sensitive to the initialization — Local optima
Spectral clustering (Hastie et al., 2009)	— Applicable to any feature type — Less sensitive to outliers, initialization and local optima — Able to identify non-convex clusters[a]	— Computationally expensive — Sensitive to the employed similarity metric
HCA[b] (Hastie et al., 2009)	— Applicable to any feature type — Less sensitive to outliers[c], initialization and local optima	— Computationally expensive — Static; divisions or fusions of clusters are irrevocable

[a]For every pair of points inside a convex cluster, the connecting straight-line segment is within this cluster.
[b]Hierarchical clustering analysis.
[c]When using the single-linkage criterion.

validation indices, we evaluate the clustering criterion with respect to the true partitioning (i.e. the actual assignment of the observations to different groups is known). Conversely, we rely on internal validation indices when we do not have the true cluster label at our disposal. With such indices, we evaluate the compactness and separation of a clustering solution. The compactness indicates how dense the clusters are and compact clusters are characterized by observations that are similar and close to each other. Clusters are well separated when observations belonging to different clusters are dissimilar and far from each other. Consequently, we employ internal validation indices to choose the value for $\mathcal{K}$ which results in compact clusters that are well separated (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019).

Several internal validation indices exist and each index formalizes the compactness and separation of the clustering solution differently. An extensive overview of internal (and external) validation indices is given by Liu et al. (2013), Wierzchoń and Kłopotek (2019), and in the benchmark study of Vendramin et al. (2010). The authors concluded that the silhouette and Caliński-Harabasz (CH) indices are superior compared to other validation criteria. While these indices are well-known within-cluster analysis (Wierzchoń and Kłopotek, 2019; Govender and Sivakumar, 2020; Vendramin et al., 2010), the results of Vendramin et al. (2010) do not necessarily generalize to our data set. We therefore include two additional, commonly used criteria: the Dunn index and Davies-Bouldin index. Table 8 shows how these four indices are calculated. For the Caliński-Harabasz, Dunn and silhouette index, higher values are associated with a better clustering solution. Conversely, for the Davies-Bouldin index, we arrive at the best partition by minimizing this criterion. A more in-depth discussion of these criteria can be found in Appendix C.

Numerous papers compared the performance of different clustering algorithms using different data sets and various evaluation criteria (Mangiameli et al., 1996; Costa et al., 2004; de Souto et al., 2008; Kinnunen et al., 2011; Jung et al., 2014; Kou et al., 2014; Rodriguez et al., 2019; Murugesan et al., 2021). They concluded that none of the considered algorithms consistently outperforms the others and that the performance is dependent on the type of data (Hennig, 2015; McNicholas, 2016a; Murugesan et al., 2021). The authors advise to compare different clustering methods using more than one performance measure. Consequently, we test the PHiRAT algorithm (see Algorithm 1) with all possible combinations of the selected clustering methods (i.e. k-medoids, spectral clustering, and HCA, see Table 7) and internal validation criteria (i.e. Caliński-Harabasz index, Davies-Bouldin index, Dunn index, and silhouette index, see Table 8).

Table 8. Internal clustering validation criteria used in this paper

Criterion	Definition
Caliński-Harabasz index (Caliński and Harabasz, 1974)	$\dfrac{\sum_{j'=1}^{J'} n_{j'} \|\boldsymbol{c}_{j'} - \boldsymbol{c}\|_2^2/(J'-1)}{\sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} \|\boldsymbol{x}_j - \boldsymbol{c}_{j'}\|_2^2/(J-J')}$ where $\boldsymbol{c} = \dfrac{1}{J}\sum_{j=1}^{J} \boldsymbol{x}_j$
Davies-Bouldin index (Davies and Bouldin, 1979)	$\dfrac{1}{J'}\sum_{j'=1}^{J'} \max_{j',\,j' \neq j'} \left(\dfrac{\frac{1}{n_{j'}}\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) + \frac{1}{n_{j'}}\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'})}{d(\boldsymbol{c}_{j'}, \boldsymbol{c}_{j'})} \right)$
Dunn index (Dunn, 1974)	$\min\limits_{1 \leqslant j' \leqslant J'} \left(\min\limits_{\substack{1 \leqslant j' \leqslant J' \\ j \neq j}} \left(\dfrac{\min\limits_{\substack{\boldsymbol{x}_j \in C_{j'} \\ \boldsymbol{x}_j \in C_{j'}}} d(\boldsymbol{x}_j, \boldsymbol{x}_j)}{\max\limits_{1 \leqslant \kappa \leqslant J'} \left\{ \max\limits_{\boldsymbol{x}_j, \boldsymbol{x}_j \in C_\kappa} d(\boldsymbol{x}_j, \boldsymbol{x}_j) \right\}} \right) \right)$
Silhouette index (Rousseeuw, 1987)	$\tilde{s} = \dfrac{1}{J}\sum_{j=1}^{J} s(\boldsymbol{x}_j)$ where $s(\boldsymbol{x}_j) = \dfrac{b(\boldsymbol{x}_j) - a(\boldsymbol{x}_j)}{\max(a(\boldsymbol{x}_j), b(\boldsymbol{x}_j))}$,
	$b(\boldsymbol{x}_j) = \min\limits_{C_{j'} \neq C_{j'}} \dfrac{1}{n_{j'}}\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{x}_j)$ and $a(\boldsymbol{x}_j) = \dfrac{1}{n_{j'}-1}\sum_{\boldsymbol{x}_j \in C_{j'},\, j \neq j} d(\boldsymbol{x}_j, \boldsymbol{x}_j)$.

$c_{j'}$ denotes the cluster center or centroid of cluster $c_{j'}$; $n_{j'}$ denotes the number of observations in cluster $c_{j'}$.

4. Clustering NACE codes in a workers' compensation insurance product

We use a workers' compensation insurance data set from a Belgian insurer to illustrate the PHiRAT procedure. The portfolio consists of Belgian companies that are active in various industries and occupations. The database contains claim-related information, such as the number of claims and the claim sizes, over a course of eight years. Additionally, for each of the companies we have the corresponding NACE-Ins code (see Section 2). In this section, we demonstrate how to reduce the NACE-Ins to its essence using PHiRAT. Our end objective is to incorporate the reduced version of the hierarchical MLF as a risk factor in a technical pricing model as discussed in Campo and Antonio (2023). We therefore also assess the effect of clustering on the predictive accuracy. Furthermore, if the reduced structure properly captures the essence, the predictive accuracy should generalize to out-of-sample and out-of-time data. We employ PHiRAT using the training data set, which contains data from the first seven years, to construct the reduced hierarchical risk factor. To examine the generalizability of the reduced structure to out-of-sample and out-of-time data, we use the test data set, which contains data from the eighth and most recent year.

4.1 Exploring the workers' compensation insurance database

4.1.1 Claim-related information at the company level

We first explore the distribution of the claim-related information in our database. We consider the damage rate Y_{it} and the number of claims N_{it} of company i in year t. When calculating $Y_{it} = \mathcal{Z}_{it}/w_{it}$ (see equation (1)), we use the capped claim amount $\mathcal{Z}_{it}$ for company i in year t to prevent that large losses have a disproportionate impact on the results (see Campo and Antonio (2023) for details on the capping procedure). To account for inflation and for the size of the company, we express the damage rate per unit of company- and year-specific salary mass w_{it}. Fig. 3 depicts the empirical distribution of the damage rates Y_{it} and number of claims N_{it} of the individual companies. A strong right skew is visible in the empirical distribution of the Y_{it} (Fig. 3(a)) and the N_{it} (Fig. 3(c)). Moreover, this right skewness persists in the log-transformed counterparts (see panels (b) and (d) in Fig. 3).
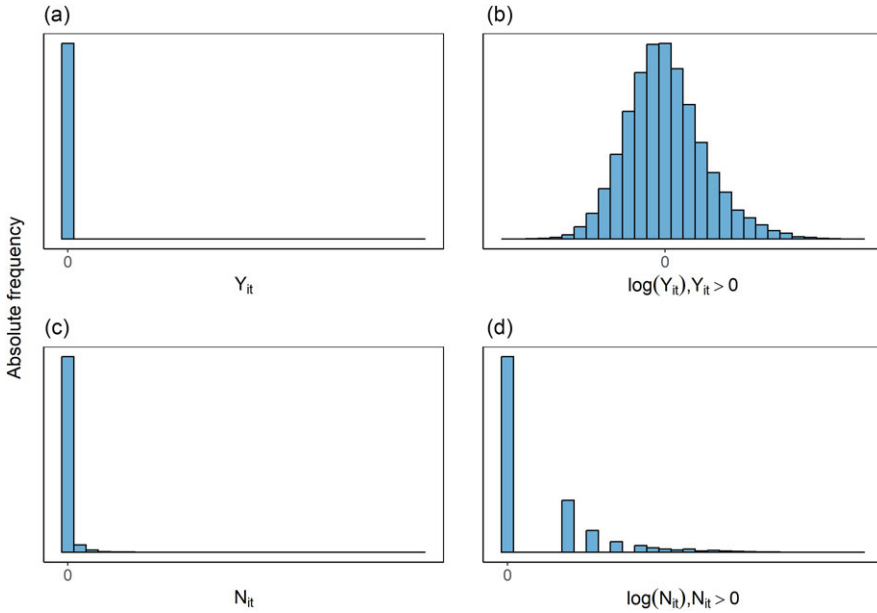
Figure 3. Empirical distribution of the individual companies' (a) damage rates Y_{it}; (b) log-transformed damage rates Y_{it} for $Y_{it} > 0$; (c) number of claims N_{it}; (d) log-transformed N_{it} for $N_{it} > 0$. This figure depicts the Y_{it}'s and N_{it}'s of all available years in our data set.

4.1.2 Claim-related information at different levels in the NACE-Ins hierarchy

The NACE-Ins partitions the companies according to their economic activity, at varying levels of granularity. We compute the category-specific weighted average damage rate and claim frequency at all levels in the hierarchy. Considering the top level in the hierarchy (as explained in Section 2), we calculate the weighted average damage rate for category $j = (1, \ldots, J)$ as

$$\bar{Y}_j = \frac{\sum_{i,k,t} w_{ijkt} Y_{ijkt}}{\sum_{i,k,t} w_{ijkt}} \tag{6}$$

and the expected claim frequency as

$$\bar{C}_j = \frac{\sum_{i,k,t} N_{ijkt}}{\sum_{i,k,t} w_{ijkt}}. \tag{7}$$

We then calculate the weighted average damage rate and expected claim frequency for child-category $jk = (j1, \ldots, jK_j)$ within parent-category j as

$$\bar{Y}_{jk} = \frac{\sum_{i,t} w_{ijkt} Y_{ijkt}}{\sum_{i,t} w_{ijkt}} \quad \text{and} \quad \bar{C}_{jk} = \frac{\sum_{i,t} N_{ijkt}}{\sum_{i,t} w_{ijkt}}. \tag{8}$$

When we consider more granular levels in the hierarchy, the computation is similar. To calculate these quantities for a specific category, we only use observations that are classified herein.

At different levels in the hierarchy, the empirical distribution of the category-specific weighted average damage rates and expected claims frequencies show a strong right skew (see Appendix D). The large range in values hinders the visual comparison of the weighted average damage rates and expected claim frequencies across categories at different levels in the hierarchy. We therefore apply
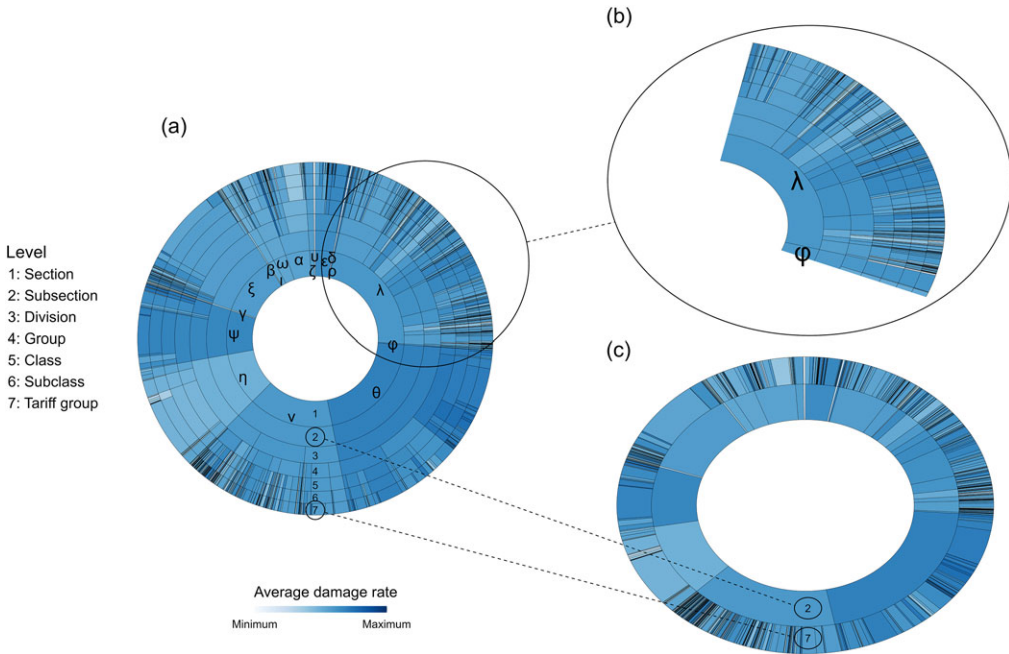
Figure 4. Category-specific weighted average damage rates: (a) at all levels in the hierarchy; (b) of the `section` λ and ϕ, including those of their child categories at all levels in the hierarchy; (c) at the `subsection` and `tariff group` level. (b) is a close-up of the top right part of (a). In this close-up, the width of ϕ at the `section` level and its child categories is increased by a factor of 10 to allow for better visual inspection.

a transformation solely for the purpose of visual comparison. Illustrating the procedure with $\bar{Y}_j$, we first apply the following transformation

$$\bar{\mathcal{y}}_j = \log(\bar{Y}_j + 0.0001) \qquad (9)$$

since we have categories for which $\bar{Y}_j = 0$. Hereafter, we cap $\bar{\mathcal{y}}_j$ using

$$\bar{\mathcal{y}}_j^c = \max(\min(\bar{\mathcal{y}}_j, \ Q_3(\bar{\mathcal{y}}_j) + 1.5 \ \mathrm{IQR}(\bar{\mathcal{y}}_j)), \ Q_1(\bar{\mathcal{y}}_j) - 1.5 \ \mathrm{IQR}(\bar{\mathcal{y}}_j)) \qquad (10)$$

where $Q_n(\bar{\mathcal{y}}_j)$ denotes the nth quantile of $\bar{\mathcal{y}}_j$ and $\mathrm{IQR}(\bar{\mathcal{y}}_j) = Q_3(\bar{\mathcal{y}}_j) - Q_1(\bar{\mathcal{y}}_j)$ denotes the interquartile range of $\bar{\mathcal{y}}_j$. The lower and upper bound of $\bar{\mathcal{y}}_j^c$ correspond to the inner fences of a boxplot (Schwertman et al., 2004). By transforming and capping the quantities, we focus on the pattern seen in the majority of the categories. Additionally, this facilitates the visual comparison across different categories.

Figure 4 visualizes the category-specific weighted average damage rates and salary masses. Panel (a) depicts the category-specific weighted averages at all levels in the hierarchy, panel (b) is close-up of the top right portion of panel (a), and panel (c) represents the averages at the `subsection` and `tariff group` level. We use a color gradient for the weighted average damage rate. The darker the color, the higher the weighted average damage rate. Further, each ring in the circle corresponds to a specific level in the hierarchy, and the level is depicted by the number in the ring. To represent the categories at a specific level, the rings are split into slices proportional to the corresponding summed salary mass. The bigger the slice, the larger the summed salary mass that corresponds to a specific category at the considered level in the hierarchy. To preserve the confidentiality of the data, we randomly assign Greek letters to each of the categories at the `section` level.
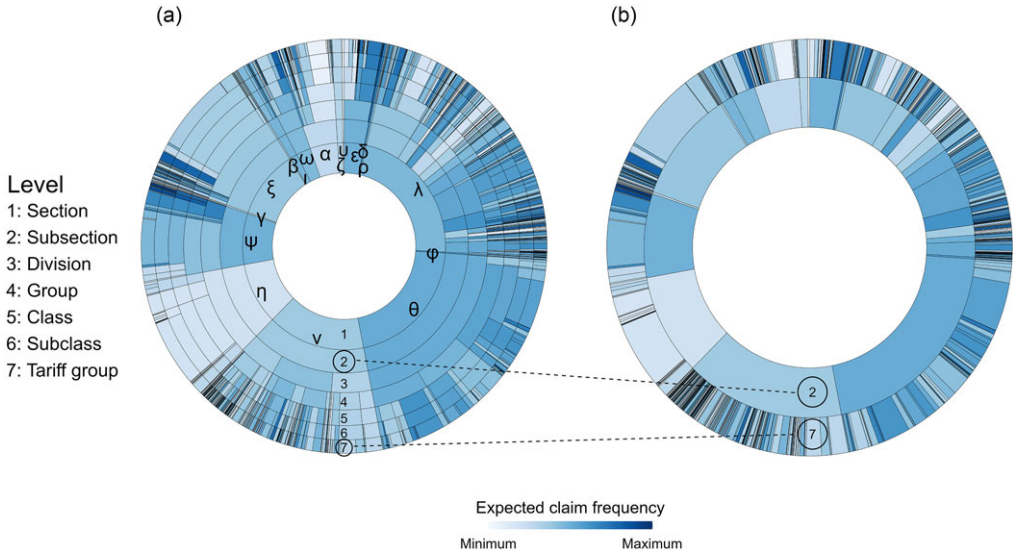
Figure 5. Category-specific expected claim frequencies: (a) at all levels in the hierarchy; (b) at the `subsection` and `tariff group` level.

At all levels in the hierarchy, there is variation in the category-specific weighted average damage rates. This is demonstrated in Fig. 4(b), displaying a magnified view of the top right section of Fig. 4(a). Here, the blue tone varies between the child categories of parent category λ at the `section` level. Additionally, at all levels in the hierarchy, there are categories with a low summed salary mass (e.g. $\gamma, \beta, \upsilon, \delta, \rho, \phi$ at the `section` level). In Fig. 4, this is depicted by the thin slices. These categories represent only a small part of our portfolio. Furthermore, most of the categories with a low salary mass have a less granular representation in the NACE system, having very few child categories compared to other categories. For example, the parent category ϕ at the `section` level has a low salary mass and only a limited number of child categories across all levels in the hierarchy of the NACE system.

Figure 5 depicts the category-specific expected claim frequency and salary masses. Similarly to Fig. 4, we use a color gradient for the claim frequencies and the size of a slice is again proportional to the summed salary mass. Overall, the findings are comparable to Fig. 4. The category-specific expected claim frequency varies between the categories at all levels in the hierarchy.

Following, we focus only on the `subsection` and `tariff group` level. The category-specific weighted average damage rates and expected claim frequencies at the `subsection` and `tariff group` level are shown in Fig. 4(c) and Fig. 5(b), respectively. To illustrate PHiRAT, we group similar categories at these levels in the hierarchy and thereby, reduce the granularity of the hierarchical risk factor.

4.2 Engineering features to improve clustering results

Feature engineering is crucial to obtain a reliable clustering solution through PHiRAT. As discussed in Section 2.2, we therefore engineer a set of features to capture the riskiness and the economic activity of the categories. The predicted random effect from the damage rate and claim frequency model expresses the category-specific riskiness at the `subsection` and `tariff group` level. Further, we use LMMs to fit the damage rate random effects models in (2) and (4). LMMs are less complex, computationally more efficient, and are less likely to experience convergence problems compared to GLMMs. Alternatively, we can consider Tweedie GLMMs to model the
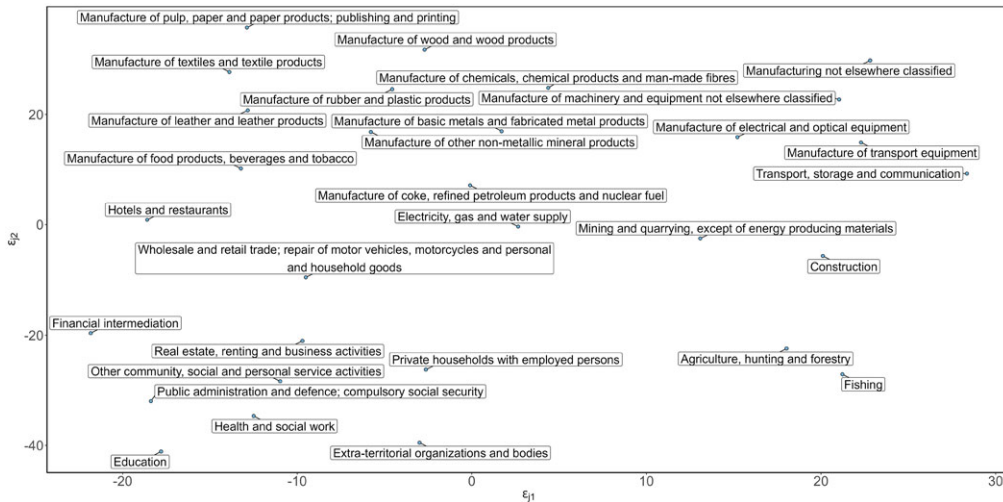
Figure 6. Low-dimensional visualization of all embedding vectors at the `subsection` level, resulting from the pre-trained Word2Vec model, encoding the textual labels of the categories (see Fig. 4). The text boxes display the textual labels. The blue dots connected to the boxes depict the position in the low-dimensional representation of the embeddings.

damage rate. We refer the reader to Campo and Antonio (2023) for a discussion on the effect of the distributional assumption on the response. We use embeddings to encode the category's textual labels that describe the economic activity. In our database, we have a description for every category at the `subsection` and `tariff group` level.

To encode the textual information, we rely on pre-trained encoders. The first pre-trained encoder we use is a Word2Vec model (Mikolov et al., 2013) trained on part of the Google News data set that contains approximately 100 billion words (https://code.google.com/archive/p/word2vec/). This encoder is only able to give vector representations of words. As a result, when encoding a sentence, for example, we get a separate vector for each word in the sentence. To obtain a single embedding for a sentence, we first remove stop words (e.g. the, and, . . .) and then take the element-wise average of the embedding vectors of the individual words (Troxler and Schelldorfer, 2022). We also use the Universal Sentence Encoder (USE) trained on Wikipedia, web news, web question-answer pages, and discussion forums (Cer et al., 2018) (https://tfhub.dev/google/collections/universal-sentence-encoder/1). There are two different versions, v4 and v5, which are specifically designed to encode greater-than-word length text (i.e. sentences, phrases, or short paragraphs). In our paper, we use both versions.

The pre-trained Word2Vec encoder outputs a 300-dimensional embedding vector and the USEs a 512-dimensional vector. To assess the quality of the resulting embeddings, we inspect whether embeddings of related economic activities lie close to each other in the vector space. We employ the dimension reduction technique t-distributed stochastic neighbor embedding (t-SNE) (Van Der Maaten and Hinton, 2008) to obtain a two-dimensional visualization of the embeddings constructed for different categories at the `subsection` level (see Fig. 5). We opt to reduce the dimensionality to two dimensions since it allows for an easy representation of the information in a scatterplot. t-SNE maps the high-dimensional embedding vector e_j for category j into a lower dimensional representation $\boldsymbol{\varepsilon}_j = (\varepsilon_{j1}, \varepsilon_{j2})$ whilst preserving its structure, enabling the visualization of relationships and the identification of patterns and groups. Figure 6 visualizes the low-dimensional representation of the embeddings resulting from the pre-trained Word2Vec encoder (see Appendix E for similar figures of USE v4 and v5).

In Fig. 6, we see that the embeddings of economically similar activities lie close to each other. All manufacturing-related activities are situated at the top of the plot and in the left bottom corner, we have mostly activities that have a social component (e.g., education). In the right bottom corner, we have activities related to the exploitation of natural resources. Further, the more unrelated the activities are, the bigger the distance between their embedding vectors. For example, *financial intermediation* ($\langle \varepsilon_{j1}, \varepsilon_{j2} \rangle \cong \langle -22.5, -20 \rangle$) and *transport, storage and communication* ($\langle \varepsilon_{j1}, \varepsilon_{j2} \rangle \cong \langle 28, 10 \rangle$) lie at opposite sides of the plot.

4.3 Clustering subsections and tariff groups using PHiRAT

We illustrate the effectiveness of PHiRAT by applying it to cluster categories at the `subsection` and `tariff group` level, using the training set. We slightly adjust Algorithm 1 to ensure that each of the resulting categories at the `subsection` and `tariff group` level has sufficient salary mass. The minimum salary mass is based on previous analyses of the insurance company and we need to adhere to this minimum during clustering. Further, as discussed at the end of Section 3, we run PHiRAT with all possible combinations of the clustering methods (i.e. k-medoids, spectral clustering, and HCA) and internal validation criteria (i.e. Caliński-Harabasz index, Davies-Bouldin index, Dunn index, and silhouette index). Per run, a specific combination is used to cluster the categories at all levels in the hierarchy.

Figure 2 visualizes how PHiRAT works its way down the hierarchy. We start at the `subsection` level and construct the feature matrix $\mathcal{F}_1$. Part of $\mathcal{F}_1$ is built using an encoder to capture the textual information (see Section 2.2 and Section 4.2). In our paper, we consider three different pre-trained encoders: (a) Word2Vec; (b) USE v4 and (c) USE v5. We use these encoders to obtain the embeddings and construct three separate feature matrices: (a) $\mathcal{F}_1^{\text{Word2Vec}}$; (b) $\mathcal{F}_1^{\text{USEv4}}$; and (c) $\mathcal{F}_1^{\text{USEv5}}$. All three feature matrices contain the same predicted random effects vectors $_1\widehat{\boldsymbol{U}}^d$ and $_1\widehat{\boldsymbol{U}}^f$. The difference between the feature matrices is that the embeddings are encoder-specific. Next, we define a tuning grid for the number of clusters $\mathcal{K}_1 \in (10, 11, \ldots, 24, 25)$. Hence, the lower and upper bound to the number of grouped categories is determined by the minimum and maximum value of the tuning grid, respectively. In combination with the encoder-specific feature matrices, this results in a search grid of three (i.e. the encoder-specific feature matrices) by 16 (i.e. the possible values for the number of clusters). For every combination in this grid, we run the clustering algorithm to obtain a clustering solution and calculate the value of the internal validation criterion. We select the combination that results in the most optimal clustering solution according to the validation measure. Figure 7 provides a visualization of this search grid. In this figure, we use k-medoids as clustering algorithm, the CH index as internal validation measure, and we use a color gradient for the value of the validation criterion. The darker the color, the higher the value and the better the clustering solution. The x-axis depicts the tuning grid for $\mathcal{K}_1$ and the y-axis the encoder-specific feature matrices. The CH index is highest (= 11.913) for $\mathcal{K}_1 = 19$ in combination with $\mathcal{F}_1^{\text{USEv5}}$. According to the selected validation measure, this specific combination results in the most optimal clustering solution. Hence, we use this clustering solution to group the categories $j = (1, \ldots, J)$ into clusters $j' = (1, \ldots, J')$ (see Fig. 2). Hereafter, we merge clusters $j' = (1, \ldots, J')$ with neighboring clusters until each cluster has sufficient salary mass.

Next, we proceed to cluster the categories at the `tariff group` level. Within each cluster j' at the `section` level, we first group consecutive categories (i.e. those with consecutive NACE codes, see Section 2.1) at the `tariff group` level to ensure that the salary mass is sufficient for every category. As before, we construct three encoder-specific versions of $\mathcal{F}_2$. We then iterate over every parent category j' to group the child categories. At iteration j', we use $\mathcal{F}_{2,\{c:\pi(c)=j'\}}$ as input in a clustering algorithm. At the `tariff group` level, we use the tuning grid $\mathcal{K}_{2,\{c:\pi(c)=j'\}} \in (5, 6, \ldots, \min(K_{j'}, 25))$. Here $K_{j'}$ denotes the total number of child categories of parent category j'. Similarly, the lower and upper bound to the number of grouped child categories is dependent on
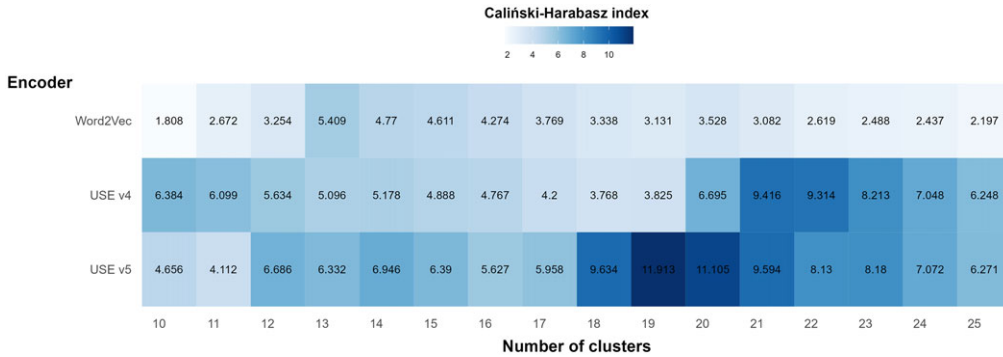
Figure 7. Visualization of the search grid. The *x*-axis depicts the tuning grid $\mathcal{K}_1$ and the *y*-axis the encoder-specific feature matrix that is used. Here, we use k-medoids for clustering and the Caliński-Harabasz (CH) index as internal validation measure. The CH index is highest (= 11.913) for $\mathcal{K}_1 = 19$ in combination with $\mathcal{F}_1^{\text{USEv5}}$, indicating that this combination results in the most optimal clustering solution.

the minimum and maximum values in the tuning grid. We select the combination of the encoder-specific feature matrix and $\mathcal{K}_{2,\{c:\pi(c)=j'\}}$ that results in the most optimal clustering solution to group the categories $j'k = (j'1, \dots, j'K_{j'})$, nested within j', into subclusters $j'k' = (j'1, \dots, j'K'_{j'})$ (see Fig. 2).

4.3.1 Distance measures and evaluation metrics

For k-medoids clustering and HCA, we use the angular distance (see Section 3.2), as both algorithms require a distance or dissimilarity measure. For spectral clustering, we use the angular similarity. We also need to define a distance measure $d(\cdot, \cdot)$ for the selected internal evaluation criteria, except for the CH index. Hereto, we define $d(\cdot, \cdot)$ as the angular distance. We calculate the evaluation criteria using all engineered features. However, since a large part of the feature vector consists of the high-dimensional embedding vector, too much weight might be given to the similarity in economic activity when choosing the optimal cluster solution. Therefore, we also evaluate a second variation of the internal evaluation criteria. When calculating the criteria, we remove the embedding vectors from the feature matrices. As such, we focus on constructing a clustering solution that is most optimal in terms of riskiness. In this evaluation, we use the Euclidean distance for $d(\cdot, \cdot)$. Hence, at the `subsection` level, we define $d(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) = \|\boldsymbol{x}_j - \boldsymbol{x}_{j'}\|_2^2$ where $\boldsymbol{x}_j = (\widehat{U}_j^d, \widehat{U}_j^f)$. Similarly, at the `tariff group` level, $d(\boldsymbol{x}_{jk}, \boldsymbol{x}_{j\ell}) = \|\boldsymbol{x}_{jk} - \boldsymbol{x}_{j\ell}\|_2^2$ and $\boldsymbol{x}_{jk} = (\widehat{U}_{jk}^d, \widehat{U}_{jk}^f)$. In what follows, we discuss the results obtained with this approach. Results obtained with the complete feature vector, including the embedding, are given in Appendix F.

4.3.2 Implementation

We perform the main part of the analysis using the statistical software R (R Core Team, 2022). For k-medoids and HCA, we rely on the `cluster` (Maechler et al., 2022) and `stats` package, respectively. For spectral clustering, we followed the implementation of Ng et al. (2001) and developed our own code. Spectral clustering partially depends on the k-means algorithm to obtain a clustering solution (see Appendix B). However, k-means is sensitive to the initialization and can get stuck in an inferior local minimum (Fränti and Sieranoja, 2019). One way to alleviate this issue is by repeating k-means with different initializations and selecting the most optimal clustering solution. Hence, to prevent spectral clustering results in a suboptimal clustering solution, we repeat the k-means step 100 times.

4.4 Evaluating the clustering solution

The main aim of our procedure is to reduce the cardinality of a hierarchically structured categorical variable, which can then be incorporated as a risk factor in a predictive model to underpin the technical price list (Campo and Antonio, 2023). Ideally, we maintain the predictive accuracy whilst reducing the granularity of the risk factor. As discussed in Section 4.3, we employ PHiRAT to group similar categories at the `subsection` and `tariff group` level. We refer to the less granular version of the hierarchical risk factor as the reduced risk factor.

To assess the predictive accuracy of a clustering solution, we fit an LMM

$$E\left[Y_{ij'k't}|U_{j'}^d, U_{j'k'}^d\right] = \mu^d + U_{j'}^d + U_{j'k'}^d \tag{11}$$

where the reduced risk factor enters the model through the random effects $U_{j'}$ and $U_{j'k'}$. We include the salary mass $w_{ij'k't}$ as weight. We opt for an LMM given its simplicity and computational efficiency (Campo and Antonio, 2023). Next, we calculate the predicted damage rate as $\widehat{Y}_{ij'k't} = \hat{\mu}^d + \widehat{U}_{j'}^d + \widehat{U}_{j'k'}^d$ for the training and test set as introduced in the beginning of Section 4.

4.4.1 Benchmark clustering solution

To evaluate the clustering solution obtained with the proposed data-driven approach, we need to compare it to a benchmark where we do not rely on PHiRAT. One possibility is to fit (11) with the nominal variable composed of the original categories at the `subsection` and `tariff group` level. In our example, fitting this LMM results in negative variance estimates and yields incorrect random effect predictions. Consequently, to obtain a benchmark clustering solution, we start at the `subsection` level and merge consecutive categories until the salary mass is sufficient (e.g. *01 agriculture* and *02 forestry*). Similarly, we index the merged categories using $j' = (1, \ldots, J')$. Hereafter, within each of the (grouped) categories at the `subsection` level, we group consecutive categories (e.g. *142121* and *142122*) at the `tariff group` level. Again, we use the minimum salary mass as defined by the insurance company and we use $k' = (1, \ldots, K'_{j'})$ as an index for the grouped categories. This results in a hierarchical MLF in which we merged adjacent categories with insufficient salary mass. To construct the benchmark model, we fit an LMM with the same specification as in (11).

4.4.2 Performance measures

Using the predicted damage rates on the training and test set, we compute the Gini index (Gini, 1921) and loss ratio. The Gini index assesses how well a model is able to distinguish high-risk from low-risk companies and is considered appropriate for the comparison of competing pricing models (Denuit et al., 2019; Campo and Antonio, 2023). The higher the value, the better the model can differentiate between risks. The Gini index has a maximum theoretical value of 1. The loss ratio measures the overall accuracy of the fitted damage rates and is defined as $\mathcal{Z}_t^{tot}/\widehat{\mathcal{Z}}_t^{tot}$, where $\mathcal{Z}_t^{tot} = \sum_{i,j',k'} \mathcal{Z}_{ij'k't}$ denotes the total capped claim amount and $\widehat{\mathcal{Z}}_t^{tot} = \sum_{i,j',k'} \widehat{\mathcal{Z}}_{ij'k't}$ the total fitted claim amount. To obtain $\widehat{\mathcal{Z}}_{ij'k't}$, we transform the individual predictions $\widehat{Y}_{ij'k't}$ as (see equation (1))

$$\widehat{\mathcal{Z}}_{ij'k't} = \widehat{Y}_{ij'k't} \cdot w_{ij'k't}. \tag{12}$$

Due to the balance property (Campo and Antonio, 2023), the loss ratio is one when calculated using the training data set. We therefore compute the loss ratio only for the test set.

Table 9. Predictive performance on the training and test set

	J'	$\sum_{j'=1}^{J'} K'_{j'}$	Training	Test	
			Gini index	Gini index	Loss ratio
Benchmark	18	641	0.658	0.585	1.006
HCA:					
Silhouette index	13	100	0.667	0.596	1.007
Dunn index	15	202	0.656	0.599	1.011
Davies-Bouldin index	14	193	0.675	0.616	1.006
CH index	15	140	0.667	0.614	1.010
k-medoids:					
Silhouette index	12	107	0.678	0.613	1.008
Dunn index	10	130	0.657	0.598	1.010
Davies-Bouldin index	14	207	0.670	0.619	1.010
CH index	10	87	0.676	0.605	1.011
Spectral clustering:					
Silhouette index	12	86	0.673	0.628	1.012
Dunn index	12	143	0.677	0.624	1.010
Davies-Bouldin index	12	166	0.669	0.618	1.010
CH index	12	102	0.670	0.628	1.013

4.4.3 Predictive performance

Table 9 depicts the performance of the different clustering solutions on the training and test sets. In this table, J' denotes the total number of grouped categories at the `subsection` level and $\sum_{j'=1}^{J'} K'_{j'}$ denotes the total number of grouped categories at the `tariff group` level. With the benchmark clustering solution, we end up with a large number of different categories at both hierarchical levels (18 + 641 = 659 separate categories in total). Conversely, with our data-driven approach, the maximum is 221 (14 + 207; k-medoids with Davies-Bouldin index) and the minimum is 97 (10 + 87; k-medoids with the CH index). Consequently, PHiRAT substantially reduces the number of categories. Moreover, we are able to retain the predictive performance. On both the training and test sets, the Gini-indices of nearly all clustering solutions are higher than the Gini index of the benchmark solution. Hence, we are better able to differentiate between high- and low-risk companies with the clustering solutions. On the training data set, the highest Gini-indices are seen for the k-medoids algorithm using the silhouette index and the spectral clustering algorithm using the Dunn index. On the test set, spectral clustering using the CH index and silhouette index result in the highest Gini index whereas the benchmark has the lowest Gini index. However, the loss ratio of most clustering solutions is higher than the loss ratio of the benchmark clustering solution (=1.006). This indicates that, when we use the reduced risk factor in an LMM, we generally underestimate the total damage. One exception is the clustering solution resulting from HCA using the Davies-Bouldin index, which has the same loss ratio as the benchmark (=1.006). In addition, for this solution, the Gini index is among the highest (0.675 on the training and 0.616 on the test set). Using the result from HCA with the Davies-Bouldin index, we are able to reduce the total number of categories to 207 (= 14 + 193). Compared to the benchmark solution, we obtain the same overall predictive accuracy (i.e. the loss ratio is approximately equal) and we can better differentiate between high- and low-risk companies.

The clustering solution resulting from HCA with the Davies-Bouldin index offers a good balance between reducing granularity and enhancing differentiation whilst preserving predictive
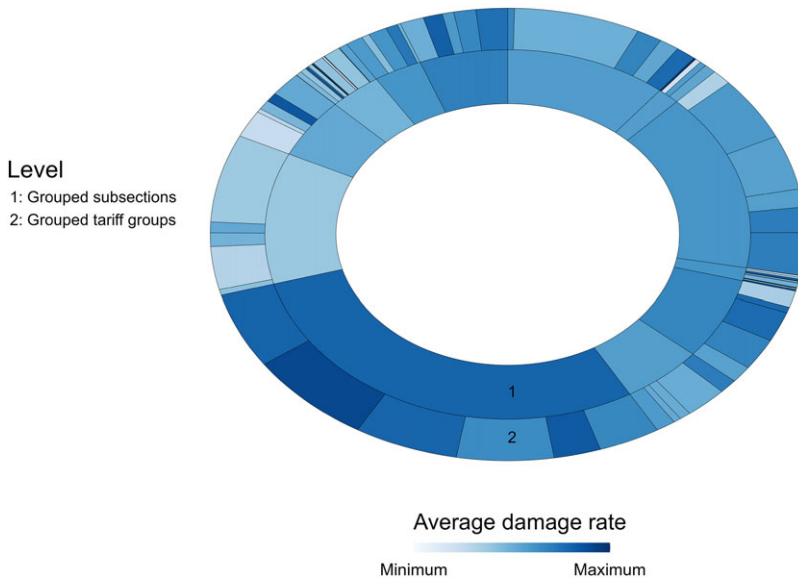
Figure 8. Cluster-specific weighted average damage rates at the `subsection` and `tariff group` level when employing PHiRAT with spectral clustering and the silhouette index.

accuracy. If, however, a sparse representation and good differentiation are more important than the overall predictive accuracy, the result from spectral clustering with the silhouette index is a better option. The latter clustering solution is visualized in Fig. 8. This figure is similar to Fig. 4(a) and depicts the cluster-specific weighted average damage rates at the `subsection` and `tariff group` level. The two figures show that PHiRAT substantially reduces the total number of categories $(12 + 86 = 98)$. Furthermore, spectral clustering with the silhouette index is one of the combinations that results in the best differentiation between high- and low-risk companies (Gini index is 0.673 and 0.628 on the training and test set, respectively).

4.4.4 Interpretability
After clustering, it is imperative to evaluate the interpretability of the solution. The grouping should provide us with meaningful insights into the underlying structure of the data. Building on the strong predictive performance of the solution resulting from spectral clustering with the silhouette index, we examine this specific clustering solution in more detail.

Figure 9 visualizes which categories are clustered at the `subsection` level. At this point, we have not yet merged neighboring clusters to ensure sufficient salary mass (see Section 4.3). Figure 9(a) shows the $\widehat{U}_j^d$ and $\widehat{U}_j^f$ of the fitted damage rate and claim frequency random effects models (see (2) and (3)). The number in the plot indicates which cluster the categories are appointed to. The description of the category's economic activity is given in Fig. 9(b). In total, we have 24 clusters, 19 of which consist of a single category. Inspecting the clusters in detail, we find that the riskiness and economic activity of grouped categories are similar. Moreover, categories that are similar in terms of riskiness (i.e. those with similar random effect predictions) but different in economic activity are assigned to different clusters. For example, the $\widehat{U}_j^d$'s and $\widehat{U}_j^f$'s are similar for: a) *manufacture of rubber and plastic products*; b) *agriculture, hunting, and forestry*; c) *manufacture of wood and wood products*; d) *manufacture of other nonmetallic mineral product,* and e) *manufacture of basic metals and fabricated metal products*. These industries are
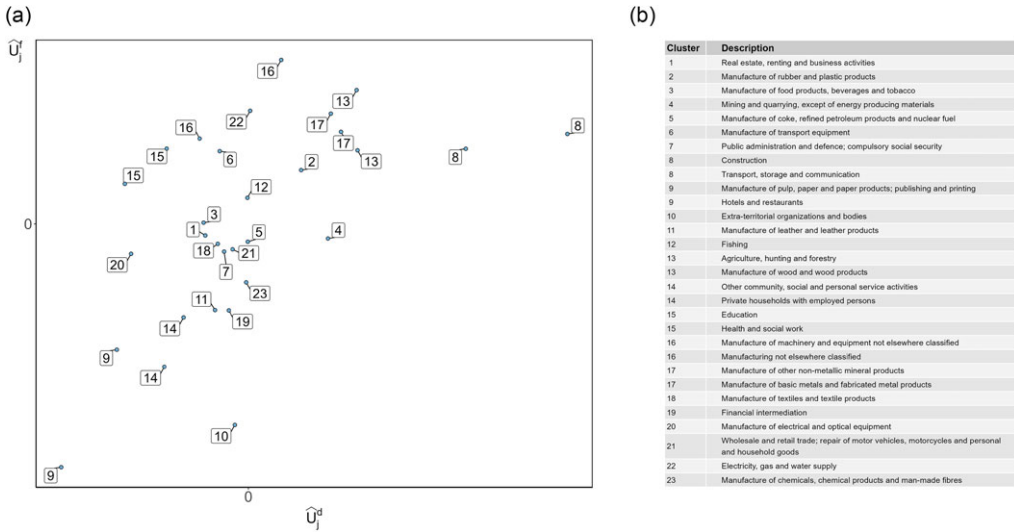
Figure 9. Visualization of the grouped categories at the `subsection` level (see Fig. 2): (a) the clustered categories and the original random effect predictions $\widehat{U}_j^d$ and $\widehat{U}_j^f$; (b) the description of the economic activity of the categories.

partitioned into three clusters. The category in cluster 2 (i.e. *manufacture of rubber and plastic products*) manufactures organic products. Conversely, the categories in cluster 17 (i.e. *manufacture of other nonmetallic mineral product* and *manufacture of basic metals and fabricated metal products*) use inorganic materials, and the categories in cluster 13 (i.e. *agriculture, hunting and forestry* and *manufacture of wood and wood products*) utilize products derived from plants and animals.

Figure 10 depicts the clustered categories k' at the `tariff group` level, nested within category $j' = $ *manufacture of chemicals, chemical products, and man-made fibers* at the `subsection` level. The $\widehat{U}_{j'k}^d$ and $\widehat{U}_{j'k}^f$ of the fitted damage rate and claim frequency random effects model (see (4) and (5)) are shown in the left panel of Fig. 10 and the textual information in the right panel. When the text is separated by a semicolon, this indicates that categories are merged before clustering to ensure sufficient salary mass (see Section 4.3). Similar to Fig. 9, both the riskiness and economic activity are taken into account when grouping categories. This is best illustrated for the categories in the red rectangle in Fig. 10(a). Herein, we have three categories: (a) *pyrotechnic items: manufacture; glue, gelatin: manufacture*; (b) *chemical basic industry* and (c) *pharmaceutical industry*. Of these, *pyrotechnic items: manufacture; glue, gelatin: manufacture* (number 7 in the top right corner of the red rectangle), and *chemical basic industry* (number 3 in the top right corner of the red rectangle) have nearly identical random effect predictions. Notwithstanding, both categories are not grouped. Instead, the category *chemical basic industry* is merged with *pharmaceutical industry* (number 3 in the bottom left corner). The latter two categories manufacture chemical compounds. In addition, companies in *chemical basic industry* mainly produce chemical compounds that are used as building blocks in other products (e.g. polymers). Building blocks (or raw materials) that are needed by companies involved in manufacturing pyrotechnic items, glue, and gelatin (e.g. glue is typically made from polymers (Ebnesajjad, 2011)).

5. Discussion

This paper presents the data-driven PHiRAT approach to reduce a hierarchically structured categorical variable with a large number of categories to its essence, by grouping similar categories
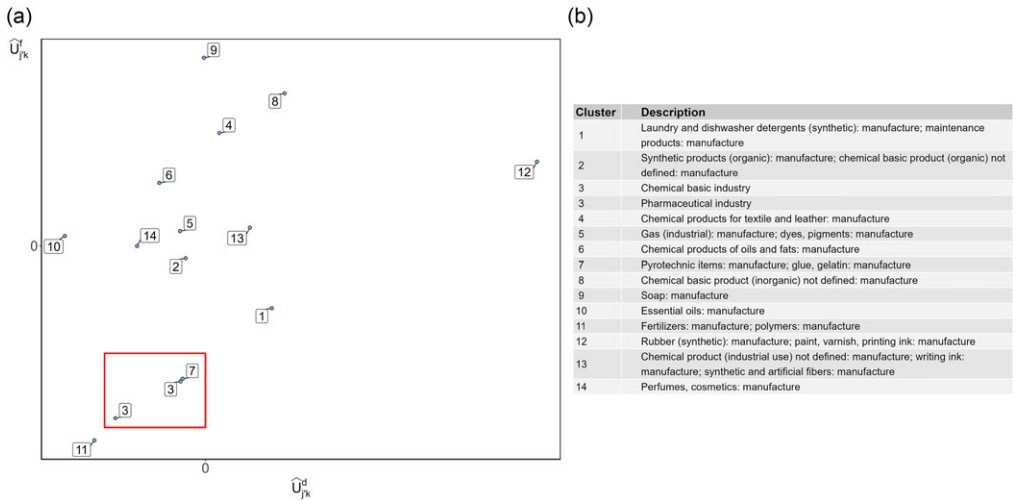
Figure 10. Visualization of the clustered child categories at the `tariff group` level, with parent category *manufacture of chemicals, chemical products, and man-made fibers* at the `subsection` level (see Fig. 2): (a) the clustered categories and the original random effect predictions $\widehat{U}^d_{j'k}$ and $\widehat{U}^f_{j'k}$; (b) the description of the economic activity of the categories.

at every level in the hierarchy. PHiRAT is a top-down procedure that preserves the hierarchical structure. It starts with grouping categories at the highest level in the hierarchy and proceeds to lower levels. At a specific level in the hierarchy, we first engineer several features to characterize the profile and specificity of each category. Using these features as input in a clustering algorithm, we group similar categories. The procedure stops once we group the categories at the lowest level in the hierarchy. When deployed in a predictive model, the reduced structure leads to a more parsimonious model that is easier to interpret, and less likely to experience estimation problems or to overfit. Further, the increased number of observations of grouped categories leads to more precise estimates of the category's effect on the response.

Using a workers' compensation insurance portfolio from a Belgian insurer, we illustrate how to employ PHiRAT to reduce the granular structure of the NACE code. Using PHiRAT, we are able to substantially reduce the dimensionality of this hierarchical risk factor whilst maintaining its predictive accuracy. The reduced risk factor allows for better differentiation, has the same overall precision as the original risk factor and the grouping seems to generalize well to out-of-sample data. Moreover, the resulting clusters consist of categories that are similar in terms of riskiness and economic activity. Furthermore, the results show that embeddings are an efficient and effective method to capture textual information.

Our approach results in a clustering solution that can provide insurers with improved insights into the underlying risk structure of a hierarchical covariate. By capturing the key characteristics of the original hierarchically structured risk factor, the reduced risk factor offers a more informative and concise representation of the various risk profiles. Insurers can incorporate the reduced version into their pricing algorithms to better assess the riskiness associated with each category in said hierarchical covariate.

Negative variance estimates in the random effects model and computational limitations with the generalized fused lasso penalty prevent us from constructing a different benchmark model. Hence, future research can examine the robustness of our approach by examining its performance in other data sets or applications and by comparing it with alternative approaches. Additionally, future research can aim to improve our proposed solution in multiple directions. The final clustering solution depends on the engineered features, the clustering algorithm, and the cluster evaluation criterion. Constructing appropriate and reliable features when employing PHiRAT is

crucial to obtain a good clustering solution. In our paper, we rely on random effects models to capture the riskiness of the categories. Alternatively, we can characterize the risk profile using entity embeddings (Guo and Berkhahn, 2016). Here, we train a neural network to create entity embeddings that map the categorical values to a continuous vector, ensuring that categories with similar response values are located closer to each other in the embedding space. Additionally, we advise to run PHiRAT using different clustering algorithms and cluster evaluation criteria, since no algorithm consistently outperforms the others (Hennig, 2015; McNicholas, 2016a,; Murugesan et al., 2021). Moreover, the robustness of the clustering solution largely depends on the stability of the selected clustering method. Further, NLP is a rapidly evolving field and we only considered three pre-trained encoders. We did not include the pre-trained Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2018), an encoder that is being increasingly used by actuarial researchers (see, for example, Xu et al. (2022); Troxler and Schelldorfer (2022)). Subsequent studies can assess whether BERT-based models result in higher-quality embeddings, which in turn leads to better clustering solutions. In addition, researchers can employ alternative approaches to represent the parent-child relationships between categories in hierarchically structured data. Such as Argyrou (2009), for example, formalizes the hierarchical relations using graph theory and applies a self-organizing map (Kohonen, 1995) to obtain a reduced representation of the hierarchical data.

Data availability statement. Due to the sensitive and confidential nature of the data used in this article, full documentation and registration of both source data and final datasets cannot be provided. The data was stored and the analyses were performed exclusively on the servers of the company providing the data.

Competing interests. No potential conflict of interest was reported by the authors.

References

Ahmad, A., **Ray**, S. K. & **Aswani Kumar**, C. (2019). Clustering mixed datasets by using similarity features. In *Sustainable Communication Networks and Application. Lecture Notes on Data Engineering and Communications Technologies* (pp. 478–485). Springer International Publishing.

Argyrou, A. (2009). Clustering hierarchical data using self-organizing map: a graph-theoretical approach. In *Advances in Self-Organizing Maps. Vol. 5629 of Lecture Notes in Computer Science* (pp. 19–27). Springer Berlin Heidelberg.

Arora, S., **May**, A., **Zhang**, J. & **Ré**, C. (2020). Contextual embeddings: when are they worth it? arXiv: 2005.09117. Available at: https://arxiv.org/abs/2005.09117

Australian Bureau of Statistics and New Zealand (2006). *Australian and New Zealand Standard Industrial Classification, (ANZSIC) 2006*. Australian Bureau of Statistics.

Breslow, N. E. & **Clayton**, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, **88**(421), 9–25.

Brown, H. & **Prescott**, R. (2006). *Applied Mixed Models in Medicine*. Wiley.

Caliński, T. & **Harabasz**, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, **3**(1), 1–27.

Campo, B. D. & **Antonio**, K. (2023). Insurance pricing with hierarchically structured data an illustration with a workers' compensation insurance portfolio. *Scandinavian Actuarial Journal*, **2023**(9), 853–884. doi: 10.1080/03461238.2022.2161413.

Carrizosa, E., **Galvis Restrepo**, M. & **Romero Morales**, D. (2021). On clustering categories of categorical predictors in generalized linear models. *Expert Systems with Applications*, **182**, 115245.

Carrizosa, E., **Mortensen**, L. H., **Romero Morales**, D. & **Sillero-Denamiel**, M. R. (2022). The tree based linear regression model for hierarchical categorical variables. *Expert Systems with Applications*, **203**, 117423.

Cer, **D.**, **Yang**, **Y.**, **yi Kong**, **S.**, **Hua**, **N.**, **Limtiaco**, **N.**, **John**, **R. S.**, **Constant**, **N.**, **Guajardo-Cespedes**, **M.**, **Yuan**, **S.**, **Tar**, **C.**, **Sung**, **Y.-H.**, **Strope**, **B.** & **Kurzweil**, **R.** (2018). Universal Sentence Encoder. arXiv: 1803.11175. Available at: https://arxiv.org/abs/1803.11175

Cheung, **Y.-m.** & **Jia**, **H.** (2013). Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognition*, **46**(8), 2228–2238.

Costa, **I. G.**, **de Carvalho**, **F.d A.** & **de Souto**, **M. C.** (2004). Comparative analysis of clustering methods for gene expression time course data. *Genetics and Molecular Biology*, **27**(4), 623–631.

Davies, **D. L.** & **Bouldin**, **D. W.** (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-1**(2), 224–227.

de Souto, **M. C.**, **Costa**, **I. G.**, **de Araujo**, **D. S.**, **Ludermir**, **T. B.** & **Schliep**, **A.** (2008). Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, **9**(1), 1–14.

Denuit, **M.**, **Sznajder**, **D.** & **Trufin**, **J.** (2019). Model selection based on Lorenz and concentration curves, Gini indices and convex order. *Insurance, Mathematics & Economics*, **89**, 128–139.

Devlin, **J.**, **Chang**, **M.-W.**, **Lee**, **K.** & **Toutanova**, **K.** (2018). BERT: pre-training of deep bidirectional transformers for language understanding, arXiv: 1810.04805. Available at: https://arxiv.org/abs/1810.04805

Dunn, **J. C.** (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, **4**(1), 95–104.

Ebnesajjad, **S.** (2011). 8 - Characteristics of adhesive materials. In S. Ebnesajjad (ed.), *Handbook of Adhesives and Surface Preparation. Plastics Design Library* (pp. 137–183). William Andrew Publishing.

European Central Bank (2021). Loans from euro area monetary financial institutions to non-financial corporations by economic activity: explanatory notes. https://www.ecb.europa.eu/stats/pdf/money/explanatory_notes_nace-en_sdw_dissemination_en.pdf?993f98fe6b628ebc6ff44b0af3d2e362.

Eurostat (2008). *NACE Rev. 2: statistical classification of economic activities in the European community*. Eurostat: Methodologies and Working Papers.

Everitt, **B.**, **Landau**, **S.** & **Leese**, **M.** (2011). *Cluster Analysis*. 5th edition, Wiley.

Ferrario, **A.** & **Naegelin**, **M.** (2020). The art of natural language processing: classical, modern and contemporary approaches to text document classification. Available at: https://ssrn.com/abstract=3547887.

FOD Economie (2004). *NACE-Bel: Activiteitennomenclatuur*. Algemene Directie Statistiek en Economische Informatie.

Foss, **A. H.**, **Markatou**, **M.** & **Ray**, **B.** (2019). Distance metrics and clustering methods for mixed-type data. *International Statistical Review*, **87**(1), 80–109.

Fränti, **P.** & **Sieranoja**, **S.** (2019). How much can k-means be improved by using better initialization and repeats? *Pattern Recognition*, **93**, 95–112.

Gelman, **A.** & **Hill**, **J.** (2017). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. 17th pr. edition. Cambridge University Press.

Gertheiss, **J.** & **Tutz**, **G.** (2010). Sparse modeling of categorial explanatory variables. *Annals of Applied Statistics*, **4**(4), 2150–2180.

Gini, **C.** (1921). Measurement of inequality of incomes. *The Economic Journal*, **31**(121), 124–126.

Govender, **P.** & **Sivakumar**, **V.** (2020). Application of k-means and hierarchical clustering techniques for analysis of air pollution: a review (1980-2019. *Atmospheric Pollution Research*, **11**(1), 40–56.

Guo, **C.** & **Berkhahn**, **F.** (2016). Entity embeddings of categorical variables. arXiv: 1604.06737. Available at: https://arxiv.org/abs/1604.06737

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

Halkidi, **M.**, **Batistakis**, **Y.** & **Vazirgiannis**, **M.** (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, **17**(2), 107–145.

Hastie, **T.**, **Tibshirani**, **R.** & **Friedman**, **J.** (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Henckaerts, **R.**, **Antonio**, **K.**, **Clijsters**, **M.** & **Verbelen**, **R.** (2018). A data driven binning strategy for the construction of insurance tariff classes. *Scandinavian Actuarial Journal*, **2018**(8), 681–705.

Hennig, **C.** (2015). What are the true clusters? *Pattern Recognition Letters*, **64**, 53–62.

Höfling, **H.**, **Binder**, **H.** & **Schumacher**, **M.** (2010). A coordinate-wise optimization algorithm for the fused lasso, arXiv: 1011.6409. Available at: https://arxiv.org/abs/1011.6409

Holizki, **T.**, **McDonald**, **R.**, **Foster**, **V.** & **Guzmicky**, **M.** (2008). Causes of work-related injuries among young workers in British Columbia. *American Journal of Industrial Medicine*, **51**(5), 357–363.

Hsu, **C.-C.** (2006). Generalizing self-organizing map for categorical data. *IEEE Transactions on Neural Networks*, **17**(2), 294–304.

Jewell, **W. S.** (1975). The use of collateral data in credibility theory : a hierarchical model. *Giornale dell'Istituto Italiano degli Attuari*, **38**, 1–16.

Jung, **Y. G.**, **Kang**, **M. S.** & **Heo**, **J.** (2014). Clustering performance comparison using k-means and expectation maximization algorithms. *Biotechnology & Biotechnological Equipment*, **28**(sup1), S44–S48.

Kaufman, **L.** & **Rousseeuw**, **P.** (1990a). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

Kaufman, **L.** & **Rousseeuw**, **P.** (1999b). *Partitioning Around Medoids (Program PAM). Chapter 2* (pp. 68–125). John Wiley & Sons, Ltd.

Kinnunen, **T.**, **Sidoroff**, **I.**, **Tuononen**, **M.** & **Fränti**, **P.** (2011). Comparison of clustering methods: a case study of text-independent speaker modeling. *Pattern Recognition Letters*, **32**(13), 1604–1617.

Kogan, **J.**, **Nicholas**, **C.** & **Teboulle**, **M.** (2005). *Grouping Multidimensional Data: Recent Advances in Clustering*. Springer Berlin/ Heidelberg.

Kohonen, **T.** (1995). *Self-organizing Maps*. Springer.

Kou, **G.**, **Peng**, **Y.** & **Wang**, **G.** (2014). Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Information Sciences*, **275**, 1–12.

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

Liu, **Y.**, **Li**, **Z.**, **Xiong**, **H.**, **Gao**, **X.**, **Wu**, **J.** & **Wu**, **S.** (2013). Understanding and enhancement of internal clustering validation measures. *IEEE Transactions on Cybernetics*, **43**(3), 982–994.

Luong, **T.**, **Socher**, **R.** & **Manning**, **C.** (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Sofia, Bulgaria (pp. 104–113). https://aclanthology.org/W13-3512

MacQueen, **J.** (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Vol 1. Oakland, CA, USA (pp. 281–297).

Maechler, **M.**, **Rousseeuw**, **P.**, **Struyf**, **A.**, **Hubert**, **M.** & **Hornik**, **K.** (2022). cluster: Cluster Analysis Basics and Extensions. R package version 2.1.4. *Available at:* https://CRAN.R-project.org/package=cluster.

Mangiameli, **P.**, **Chen**, **S. K.** & **West**, **D.** (1996). A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research*, **93**(2), 402–417.

McNicholas, **P.** (2016a). *Mixture Model-Based Classification*. CRC Press.

McNicholas, **P. D.** (2016b). Model-based clustering. *Journal of Classification*, **33**(3), 331–373.

Micci-Barreca, **D.** (2001). A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *SIGKDD Explorations*, **3**(1), 27–32.

Mikolov, **T.**, **Chen**, **K.**, **Corrado**, **G.** & **Dean**, **J.** (2013). Efficient estimation of word representations in vector space, arXiv: 1301.3781. Available at: https://arxiv.org/abs/1301.3781

Mohammad, **S. M.** & **Hirst**, **G.** (2012). Distributional measures of semantic distance: a survey, arXiv: 1203.1858. Available at: https://arxiv.org/abs/1203.1858

Molenberghs, **G.** & **Verbeke**, **G.** (2005). *Models for Discrete Longitudinal Data*. Springer New York.

Molenberghs, **G.** & **Verbeke**, **G.** (2011). A note on a hierarchical interpretation for negative variance components. *Statistical Modelling*, **11**(5), 389–408.

Murugesan, **N.**, **Cho**, **I.** & **Tortora**, **C.** (2021). Benchmarking in cluster analysis: a study on spectral clustering, dbscan, and k-means. In *Data Analysis and Rationality in a Complex World. Vol. 5 of Studies in Classification, Data Analysis, and Knowledge Organization* (pp. 175–185). Springer International Publishing.

Ng, **A.**, **Jordan**, **M.** & **Weiss**, **Y.** (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, **14**.

Oelker, **M.-R.**, **Gertheiss**, **J.** & **Tutz**, **G.** (2014). Regularization and model selection with categorical predictors and effect modifiers in generalized linear models. *Statistical Modelling*, **14**(2), 157–177.

Ohlsson, **E.** (2008). Combining generalized linear models and credibility models in practice. *Scandinavian Actuarial Journal*, **2008**(4), 301–314.

Ohlsson, **E.** & **Johansson**, **B.** (2010). *Non-Life Insurance Pricing with Generalized Linear Models*. Springer Berlin Heidelberg: Imprint: Springer.

Oliveira, **I.**, **Molenberghs**, **G.**, **Verbeke**, **G.**, **Demetrio**, **C.** & **Dias**, **C.** (2017). Negative variance components for non-negative hierarchical data with correlation, over-, and/or underdispersion. *Journal of Applied Statistics*, **44**(6), 1047–1063.

Onan, **A.** (2017). A k-medoids based clustering scheme with an application to document clustering. In *2017 International Conference on Computer Science and Engineering (UBMK)* (pp. 354–359).

Ostrovsky, **R.**, **Rabani**, **Y.**, **Schulman**, **L.** & **Swamy**, **C.** (2012). The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM*, **59**(6), 1–22.

Pargent, **F.**, **Pfisterer**, **F.**, **Thomas**, **J.** & **Bischl**, **B.** (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics*, **37**(5), 2671–2692.

Phillips, **J. M.** (2021). *Mathematical Foundations for Data Analysis*. Springer International Publishing.

Pinheiro, **J.**, **Pinheiro**, **J.** & **Bates**, **D.** (2009). *Mixed-Effects Models in S and S-PLUS*. Springer.

Poon, **L. K. M.**, **Liu**, **A. H.**, **Liu**, **T.** & **Zhang**, **N. L.** (2012). A model-based approach to rounding in spectral clustering, arXiv: 1210.4883. Available at: https://arxiv.org/abs/1210.4883

Pryseley, **A.**, **Tchonlafi**, **C.**, **Verbeke**, **G.** & **Molenberghs**, **G.** (2011). Estimating negative variance components from gaussian and non-gaussian data: a mixed models approach. *Computational Statistics & Data Analysis*, **55**(2), 1071–1085.

R Core Team. (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. https://www.R-project.org/.

Rentzmann, **S.** & **Wuthrich**, **M. V.** (2019). Unsupervised learning: what is a sports car? Available at: https://ssrn.com/abstract=3439358 or 10.2139/ssrn.3439358.

Rodriguez, **M. Z.**, **Comin**, **C. H.**, **Casanova**, **D.**, **Bruno**, **O. M.**, **Amancio**, **D. R.**, **Costa**, **L.d F.** & **Rodrigues**, **F. A.** (2019). Clustering algorithms: a comparative approach. *PloS One*, **14**(1), e0210236.

Rosenberg, **M.** & **Zhong**, **F.** (2022). Using clusters based on social determinants to identify the top 5% utilizers of health care. *North American Actuarial Journal*, **26**(3), 456–469.

Rousseeuw, **P. J.** (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, **20**, 53–65.

Schomacker, **T.** & **Tropmann-Frick**, **M.** (2021). Language representation models: an overview. *Entropy (Basel, Switzerland)*, **23**(11), 1422.

Schubert, **E.** (2021). A triangle inequality for cosine similarity. In *Similarity Search and Applications. Lecture Notes in Computer Science* (pp. 32–44). Springer International Publishing.

Schwertman, **N. C.**, **Owens**, **M. A.** & **Adnan**, **R.** (2004). A simple more general boxplot method for identifying outliers. *Computational Statistics & Data Analysis*, **47**(1), 165–174.

Stassen, **B.**, **Denuit**, **M.**, **Mahy**, **S.**, **Maréchal**, **X.** & **Trufin**, **J.** (2017). A unified approach for the modelling of rating factors in workers compensation insurance. White paper. Reacfin. *Available at:* https://www.reacfin.com/wp-content/uploads/2016/12/170131-Reacfin-White-Paper-A-Unified-Approach-for-the-Modeling-of-Rating-Factors-in-Work-ers%E2%80%99-Compensation-Insurance.pdf.

Statistical Office of the European Communities (1996). *NACE Rev. 1: Statistical Classification of Economic Activities in the European Community*. Office for Official Publications of the European Communities.

Struyf, **A.**, **Hubert**, **M.** & **Rousseeuw**, **P.** (1997). Clustering in an object-oriented environment. *Journal of Statistical Software*, **1**(1), 1–30.

Timm, **N. H.** (2002). *Applied Multivariate Analysis*. Springer.

Troxler, **A.** & **Schelldorfer**, **J.** (2022). Actuarial applications of natural language processing using transformers: case studies for using text features in an actuarial context, arXiv: 2206.02014. Available at: https://arxiv.org/abs/2206.02014

Tutz, **G.** & **Oelker**, **M.** (2017). Modelling clustered heterogeneity: fixed effects, random effects and mixtures. *International Statistical Review*, **85**(2), 204–227.

Van Der Maaten, **L.** & **Hinton**, **G.** (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2625.

Vendramin, **L.**, **Campello**, **R. J. G. B.** & **Hruschka**, **E. R.** (2010). Relative clustering validity criteria: a comparative overview. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **3**(4), 209–235.

Verma, **D.** & **Meila**, **M.** (2003). *A Comparison of Spectral Clustering Algorithms. Technical Report*. University of Washington Tech Rep UWCSE030501.

Verma, **V. K.**, **Pandey**, **M.**, **Jain**, **T.** & **Tiwari**, **P. K.** (2021). Dissecting word embeddings and language models in natural language processing. *Journal of Discrete Mathematical Sciences & Cryptography*, **24**(5), 1509–1515.

von Luxburg, **U.** (2007). A tutorial on spectral clustering, arXiv: 0711.0189. Available at: https://arxiv.org/abs/0711.0189

von Luxburg, **U.**, **Belkin**, **M.** & **Bousquet**, **O.** (2008). Consistency of spectral clustering. *The Annals of Statistics*, **36**(2), 555–586.

von Luxburg, **U.**, **Bousquet**, **O.** & **Belkin**, **M.** (2004). On the convergence of spectral clustering on random samples: The normalized case. In 'LEARNING THEORY, PROCEEDINGS'. Vol. 3120 of Lecture Notes in Computer Science (pp. 457–471). Springer Berlin Heidelberg.

Walters, **J. K.**, **A. Christensen**, **K.**, **K. Green**, **M.**, **E. Karam**, **L.**, **D. Kincl**, **L.** (2010). Occupational injuries to oregon workers 24 years and younger: an analysis of workers' compensation claims, 2000-2007. *American Journal of Industrial Medicine*, **53**(10), 984–994.

Wang, **X.** & **Keogh**, **E.** (2008). A clustering analysis for target group identification by locality in motor insurance industry. In *'Soft Computing Applications in Business'. Vol. 230 of Studies in Fuzziness and Soft Computing* (pp. 113–127). Springer Berlin Heidelberg.

Wierzchoń, **S.** & **Kłopotek**, **M.** (2019). *Modern Algorithms of Cluster Analysis*. Springer International Publishing.

Wurzelbacher, **S. J.**, **Meyers**, **A. R.**, **Lampl**, **M. P.**, **Timothy Bushnell**, **P.**, **Bertke**, **S. J.**, **Robins**, **D. C.**, **Tseng**, **C.-Y.** & **Naber**, **S. J.** (2021). Workers' compensation claim counts and rates by injury event/exposure among state-insured private employers in ohio, 2007-2017. *Journal of Safety Research*, **79**, 148–167.

Wüthrich, **M. V.** (2017). Covariate selection from telematics car driving data. *European Actuarial Journal*, **7**(1), 89–108.

Xu, **S.**, **Zhang**, **C.** & **Hong**, **D.** (2022). BERT-based NLP techniques for classification and severity modeling in basic warranty data study. *Insurance: Mathematics and Economics*, **107**, 57–67.

Yeo, **A. C.**, **Smith**, **K. A.**, **Willis**, **R. J.** & **Brooks**, **M.** (2001). Clustering technique for risk classification and prediction of claim costs in the automobile insurance industry. *Intelligent Systems in Accounting, Finance and Management*, **10**(1), 39–50.

Yu, **D.**, **Liu**, **G.**, **Guo**, **M.** & **Liu**, **X.** (2018). An improved k-medoids algorithm based on step increasing and optimizing medoids. *Expert Systems with Applications*, **92**, 464–473.

Zappa, D., **Borrelli**, M., **Clemente**, G. P. & **Savelli**, N. (2021). Text mining in insurance: from unstructured data to meaning. *Variance*, **14**(1), 1–15.

Zhu, R. & **Wüthrich**, M. V. (2021). Clustering driving styles via image processing. *Annals of Actuarial Science*, **15**(2), 276–290.

Appendix A. Distance and (dis)similarity metrics

Using clustering algorithms, we aim to divide a set of data points $x_1, \ldots, x_J$ into J' homogeneous groups such that observations in each cluster j' are more similar to each other compared to observations of other clusters $j' \neq j'$. Consequently, most clustering algorithms rely on distance or (dis)similarity metrics between all pairwise observations.

The most commonly used distance metric between two vectors x_j and $x_{\tilde{j}}$ is the squared Euclidean distance

$$d_e(x_j, x_{\tilde{j}}) = \|x_j - x_{\tilde{j}}\|_2^2. \tag{13}$$

Here, $\|x_j\|_2 := \sqrt{x_{j1}^2 + \cdots + x_{jn_f}^2}$ and n_f denotes the number of features considered. The squared Euclidean distance can be converted to the Gaussian similarity measure

$$s_g(x_j, x_{\tilde{j}}) = \exp\left(\frac{-\|x_j - x_{\tilde{j}}\|_2^2}{\sigma^2}\right) \tag{14}$$

which ranges from 0 (i.e. dissimilar) to 1 (i.e. identical). σ is a scaling parameter set by the user (Ng et al., 2001; Poon et al., 2012). When σ is small, the distance needs to be close to 0 to result in a high similarity measure. Conversely, for high σ, even large distances will result in a value close to 1.

Ideally, vectors that lie close to each other are characterized by a low distance $d(\cdot, \cdot)$ and a high similarity measure $s(\cdot, \cdot)$. Euclidean-based distance/similarity measures, however, are not appropriate to capture the similarities between embeddings (Kogan et al., 2005). Within NLP, the cosine similarity

$$s_c(x_j, x_{\tilde{j}}) = \frac{x_j^\top x_{\tilde{j}}}{\|x_j\|_2 \cdot \|x_{\tilde{j}}\|_2} \tag{15}$$

is therefore most often used to measure the similarity between embeddings (Mohammad and Hirst, 2012; Schubert, 2021). The cosine similarity ranges from −1 (opposite) to 1 (similar). In cluster analysis, however, we generally require the (dis)similarity measure to range from 0 to 1 (Everitt et al., 2011; Kogan et al., 2005; Hastie et al., 2009). In this case, we can use the angular similarity

$$s_a(x_j, x_{\tilde{j}}) = 1 - \frac{\cos^{-1}(s_c(x_j, x_{\tilde{j}}))}{\pi} \tag{16}$$

which is restricted to [0, 1]. Hereto related is the angular distance

$$d_a(x_j, x_{\tilde{j}}) = \frac{\cos^{-1}(s_c(x_j, x_{\tilde{j}}))}{\pi}. \tag{17}$$

The angular distance is a proper distance metric since it satisfies the triangle inequality $d(x_j, x_{\tilde{j}}) \leqslant d(x_j, x_z) + d(x_z, x_{\tilde{j}})$ for any z (Schubert, 2021; Phillips, 2021). Conversely, the distance measure based on the cosine similarity does not satisfy this inequality.

Appendix B. Clustering algorithms

B.1 K-means clustering

With k-means clustering (MacQueen et al., 1967), we group the J categories into J' clusters $(C_1, \ldots, C_{J'})$ by minimizing

$$\underset{(C_1,\ldots,C_{J'})}{\arg\min} \sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) = \underset{(C_1,\ldots,C_{J'})}{\arg\min} \sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} \|\boldsymbol{x}_j - \boldsymbol{c}_{j'}\|_2^2 \qquad (18)$$

where $\boldsymbol{c}_{j'}$ denotes the cluster center or centroid of cluster $C_{j'}$. $\boldsymbol{c}_{j'}$ is the sample mean of all $\boldsymbol{x}_j \in C_{j'}$

$$\boldsymbol{c}_{j'} = \frac{1}{n_{j'}} \sum_{\boldsymbol{x}_j \in C_{j'}} \boldsymbol{x}_j. \qquad (19)$$

where $n_{j'}$ denotes the number of observations in cluster $C_{j'}$. Hence, with (18) we minimize the within-cluster sum of squares.

K-means is only suited for numeric features, is sensitive to outliers, has several local optima and the results are sensitive to the initialization (Kogan et al., 2005; Everitt et al., 2011; Ostrovsky et al., 2012; Hastie et al., 2009).

B.2 K-medoids clustering

Contrary to k-means, k-medoids clustering (Kaufman and Rousseeuw, 1990b) uses an existing data point $\boldsymbol{x}_j$ as cluster center. In addition, the distance measure in k-medoids clustering is not restricted to the Euclidean distance (Rentzmann and Wuthrich, 2019; Hastie et al., 2009). It can be used with any distance or dissimilarity measure. With k-medoids clustering we minimize

$$\underset{(C_1,\ldots,C_{J'})}{\arg\min} \sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) \qquad (20)$$

where $\boldsymbol{c}_{j'}$ is the observation for which $\sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'})$ is minimal (Struyf et al., 1997). This observation is most central within cluster $C_{j'}$ and is called a medoid.

K-medoids is applicable to any feature type and is less sensitive to outliers. Nonetheless, it still suffers from local optima and is sensitive to the initialization (Onan, 2017; Yu et al., 2018).

B.3 Spectral clustering

In spectral clustering, we represent the data using an undirected similarity graph $G = \langle V, E \rangle$, where $V = (v_1, \ldots, v_j, \ldots, v_J)$ stands for the set of vertices and E denotes the set of edges (Hastie et al., 2009; von Luxburg, 2007; Wierzchoń and Kłopotek, 2019). The weight of the edges are represented using a $J \times J$ similarity matrix $\mathcal{S}$ which contains all pairwise similarities $s(\cdot, \cdot) \geqslant 0$ between the observations. The diagonal entries in the $\mathcal{S}$ matrix are equal to zero. Vertices v_j and $v_{\tilde{j}}$ are connected if $s(\boldsymbol{x}_j, \boldsymbol{x}_{\tilde{j}}) > 0$. Hereby, we reformulate clustering as a graph-partitioning problem. We want to partition the graph such that edges within a group j' have high weights and edges between different groups $j' \neq \tilde{j}'$ have low weights.

To represent the degree of the vertices, we set up a diagonal matrix $\mathcal{D}$ with diagonal elements $(j, j) = \sum_{\tilde{j}=1}^{J} s(\boldsymbol{x}_j, \boldsymbol{x}_{\tilde{j}})$. We use $\mathcal{D}$ to transform the similarity matrix to the Laplacian matrix $\mathcal{L} = \mathcal{D} - \mathcal{S}$. Next, we compute the J eigenvectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_J)$ of $\mathcal{L}$. To cluster the observations in J' groups, we use the J' eigenvectors $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{J'})$ corresponding to the smallest eigenvalues and

stack these in columns to form the matrix $U \in \mathbb{R}^{J \times J'}$. In U, the jth row corresponds to the original observation x_j. In the ideal case of J' true clusters, $\mathcal{L}$ is a block diagonal matrix

$$\mathcal{L} = \begin{bmatrix} \mathcal{L}_1 & & & \\ & \mathcal{L}_2 & & \\ & & \ddots & \\ & & & \mathcal{L}_{J'} \end{bmatrix} \tag{21}$$

when the vertices are ordered according to their cluster membership. Here, $\mathcal{L}_{j'}$ is the block corresponding to cluster j'. In this situation, $\mathcal{L}$ has J' eigenvectors with eigenvalue zero and these eigenvectors are indicator vectors (i.e. the values are 1 for a specific cluster j' and 0 for clusters $j'' \neq j'$) (Hastie et al., 2009; Poon et al., 2012; von Luxburg, 2007). This allows us to easily identify the J' groups. Consequently, in a second step, we apply k-means to U and the results hereof determine the clustering solution.

We commonly refer to $\mathcal{L}$ as the unnormalized Laplacian. It is, however, preferred to use a normalized version of $\mathcal{L}$ (von Luxburg et al., 2004; von Luxburg et al., 2008). One way to normalize $\mathcal{L}$ is by applying the following transformation to $\mathcal{L}$

$$\mathcal{D}^{-1/2} \mathcal{L} \mathcal{D}^{-1/2} = I - \mathcal{D}^{-1/2} \mathcal{S} \mathcal{D}^{-1/2}. \tag{22}$$

Spectral clustering can be used with any feature type and is less sensitive to initialization issues, outliers, and local optima (Verma and Meila, 2003; von Luxburg, 2007). Moreover, spectral clustering is specifically designed to identify non-convex clusters (for every pair of points inside a convex cluster, the connecting straight-line segment is within this cluster). Conversely, k-means, k-medoids, and HCA generally do not work well with non-convex clusters (Hastie et al., 2009; von Luxburg, 2007). When compared to other clustering algorithms, spectral clustering often has a better overall performance (Murugesan et al., 2021; Rodriguez et al., 2019). Notwithstanding, spectral clustering is sensitive to the employed similarity metric (Haberman and Renshaw, 1996; von Luxburg, 2007; de Souto et al., 2008).

B.4 Hierarchical clustering analysis

Contrary to the other clustering methods, hierarchical clustering analysis (HCA) does not start from a specification of the number of clusters. Instead, it builds a hierarchy of clusters which can be either top-down or bottom-up (Hastie et al., 2009). In the agglomerative or bottom-up approach, each observation is initially assigned to its own cluster and we recursively merge clusters into a single cluster. When merging two clusters, we select the pair for which the dissimilarity is smallest. Conversely, in the divisive or top-down approach, we start with all observations assigned to one cluster and at each step, the algorithm recursively splits one of the existing clusters into two new clusters. Here, we split the cluster resulting in the largest between-cluster dissimilarity. Consequently, in both approaches, we need to define how to measure the dissimilarity between two clusters. With complete-linkage, for example, the distance between two clusters $C_{j'}$ and $C_{j'}$ is defined as the maximum distance $d(\cdot, \cdot)$ between two observations in the separate clusters

$$d_{CL}(C_{j'}, C_{j'}) = \max_{\substack{x_j \in C_{j'}, \\ x_{\hat{j}} \in C_{j'}}} (d(x_j, x_{\hat{j}})). \tag{23}$$

Conversely, with single-linkage, we define the distance between two clusters as

$$d_{SL}(C_{j'}, C_{j'}) = \min_{\substack{x_j \in C_{j'}, \\ x_j \in C_{j'}}} (d(x_j, x_j)). \tag{24}$$

Several other methods are available and we refer the reader to Hastie et al. (2009) for an overview. The visualization of the different steps in HCA is often referred to as a dendrogram. It plots a tree-like structure which shows how the clusters are formed at each step in the algorithm. To partition the data into J' clusters, we cut the dendrogram horizontally at the height that results in J' clusters.

Due to its design, HCA is less sensitive to initialization issues and local optima in comparison to k-means. In addition, we can employ HCA with any type of feature, and HCA with single linkage is more robust to outliers (Everitt et al., 2011; Timm, 2002). The disadvantage of HCA is that divisions or fusions of clusters are irrevocable (Kaufman and Rousseeuw, 1990a; Kogan et al., 2005). Once a cluster has been split or merged, it cannot be undone.

Appendix C. Internal cluster evaluation criteria

In the aforementioned clustering techniques, J' can be considered a tuning parameter that needs to be carefully chosen from a range of different (integer) values. Hereto, we select the value of J' that results in the most optimal clustering solution according to a chosen cluster validation index. We divide the cluster validation indices into two groups, internal and external (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019; Halkidi et al., 2001). Using external validation indices, we evaluate the clustering criterion with respect to the true partitioning (i.e. the actual assignment of the observations to different groups is known). Conversely, we rely on internal validation indices when we do not have the true cluster label at our disposal. Here, we evaluate the compactness and separation of a clustering solution. The compactness indicates how dense the clusters are and compact clusters are characterized by observations that are similar and close to each other. Clusters are well separated when observations of different clusters are dissimilar and far from each other. Consequently, we employ internal validation indices to choose that J' which results in compact clusters that are well separated (Liu et al., 2013; Everitt et al., 2011; Wierzchoń and Kłopotek, 2019).

Several internal validation indices exist and each index formalizes the compactness and separation of the clustering solution differently. An extensive overview of internal (and external) validation indices is given by Liu et al. (2013) and Wierzchoń and Kłopotek (2019). Vendramin et al. (2010) conducted an extensive comparison of the performance of 40 internal validation criteria using 1080 data sets. These data sets were grouped into four categories: a) a low number of features (i.e. $\in (2, 3, 4)$); b) a high number of features (i.e. $\in (22, 23, 24)$); c) a low number of true clusters (i.e. $\in (2, 4, 6)$) and d) a high number of true clusters (i.e. $\in (12, 14, 16)$). The authors concluded that the silhouette and Caliński-Harabasz indices are superior compared to other validation criteria. These indices are well-known within-cluster analysis (Wierzchoń and Kłopotek, 2019; Govender and Sivakumar, 2020; Vendramin et al., 2010). Nonetheless, the results of Vendramin et al. (2010) do not necessarily generalize to our data set. We therefore include two additional, commonly used criteria: the Dunn index and Davies-Bouldin index.

C.1 Caliński-Harabasz index

The Caliński-Harabasz (CH) index (Caliński and Harabasz, 1974; Liu et al., 2013) is defined as the ratio of the average between- to the within-sum of squares

$$\frac{\sum_{j'=1}^{J'} n_{j'} \|\boldsymbol{c}_{j'} - \boldsymbol{c}\|_2^2 / (J' - 1)}{\sum_{j'=1}^{J'} \sum_{\boldsymbol{x}_j \in C_{j'}} \|\boldsymbol{x}_j - \boldsymbol{c}_{j'}\|_2^2 / (J - J')} \tag{25}$$

where $\boldsymbol{c}$ denotes the global center of all observations. We compute $\boldsymbol{c}$ as

$$\boldsymbol{c} = \frac{1}{J} \sum_{j=1}^{J} \boldsymbol{x}_j. \tag{26}$$

The higher this index, the more compact and well-separated the clustering solution. This index is also known as the Pseudo F-statistic. The results of Vendramin et al. (2010) suggest that the CH index performs better when the number of features is high and the number of true clusters is low.

C.2 Davies-Bouldin index

The Davis-Bouldin index is defined as (Davies and Bouldin, 1979)

$$\frac{1}{J'} \sum_{j'=1}^{J'} \max_{j', \, j' \neq j'} \left(\frac{\frac{1}{n_{j'}} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'}) + \frac{1}{n_{j'}} \sum_{\boldsymbol{x}_j \in C_{j'}} d(\boldsymbol{x}_j, \boldsymbol{c}_{j'})}{d(\boldsymbol{c}_{j'}, \boldsymbol{c}_{j'})} \right). \tag{27}$$

The numerator in (27) captures the compactness of clusters $C_{j'}$ and $C_{j'}$ and dense clusters are characterized by low values. With the denominator, we measure the distance between the centroids of clusters $C_{j'}$ and $C_{j'}$ and this signifies how well separated the clusters are. Hence, low ratios are indicative of dense clusters that are well separated. By taking the maximum ratio for a specific cluster $C_{j'}$ we take the worst scenario possible.

According to Vendramin et al. (2010), the Davis-Bouldin index performs better for data sets with fewer features and this finding was more pronounced for data sets with a low number of true clusters.

C.3 Dunn index

The Dunn index (Dunn, 1974) is defined as the ratio of the minimum distance between the clusters to the maximum distance within clusters

$$\min_{1 \leqslant j' \leqslant J'} \left(\min_{\substack{1 \leqslant j' \leqslant J' \\ j \neq j}} \left(\frac{\min_{\substack{x_j \in C_{j'} \\ x_j \in C_{j'}}} d(x_j, x_j)}{\max_{1 \leqslant \kappa \leqslant J'} \left\{ \max_{x_j, x_j \in C_\kappa} d(x_j, x_j) \right\}} \right) \right) \tag{28}$$

The higher this index, the better the clustering solution. Several variants exist of the Dunn index (see, for example, Vendramin et al. (2010)) and we focus on the original formulation as given in (28). In Vendramin et al. (2010), the Dunn index performed reasonably when focusing on the difference between the true and selected number of clusters. Notwithstanding, of all four indices considered in our paper, it has the lowest performance.

C.4 Silhouette index

For a specific observation $\boldsymbol{x}_j \in C_{j'}$, we define the average dissimilarity of $\boldsymbol{x}_j$ to all other observations in cluster $C_{j'}$ as

$$a(\boldsymbol{x}_j) = \frac{1}{n_{j'} - 1} \sum_{\boldsymbol{x}_{\hat{j}} \in C_{j'}, \hat{j} \neq j} d(x_j, x_{\hat{j}}) \tag{29}$$

and the average dissimilarity of $\boldsymbol{x}_j$ to all observations in cluster $C_{\hat{j}'}$ as

$$e(\boldsymbol{x}_j) = \frac{1}{n_{\hat{j}'}} \sum_{\boldsymbol{x}_{\hat{j}} \in C_{\hat{j}'}} d(\boldsymbol{x}_j, \boldsymbol{x}_{\hat{j}}). \tag{30}$$

We compute $e(\boldsymbol{x}_j)$ for all clusters $C_{j'} \neq C_{\hat{j}'}$ and calculate

$$b(\boldsymbol{x}_j) = \min_{C_{j'} \neq C_{\hat{j}'}} e(\boldsymbol{x}_j). \tag{31}$$

We call $b(\boldsymbol{x}_j)$ the neighbor of $\boldsymbol{x}_j$ as it is the closest observation of another cluster. We calculate the silhouette value $s(\boldsymbol{x}_j)$ as

$$s(\boldsymbol{x}_j) = \frac{b(\boldsymbol{x}_j) - a(\boldsymbol{x}_j)}{\max(a(\boldsymbol{x}_j), b(\boldsymbol{x}_j))}. \tag{32}$$

$s(\boldsymbol{x}_j)$ indicates how well an observation $\boldsymbol{x}_j$ is clustered and from the definition, it follows that $-1 \leqslant s(\boldsymbol{x}_j) \leqslant 1$. For clusters with a single observation, we set $s(\cdot) = 0$. Values close to one indicate that the observation has been assigned to the appropriate cluster, since the smallest between dissimilarity $b(\boldsymbol{x}_j)$ is much larger than the within dissimilarity $a(\boldsymbol{x}_j)$ (Rousseeuw, 1987). Conversely, when $s(\boldsymbol{x}_j)$ is close to -1, $\boldsymbol{x}_j$ lies on average closer to the neighboring cluster than to its own cluster and this suggests that this observation is not assigned to the appropriate cluster. We calculate the average silhouette width

$$\tilde{s} = \frac{1}{J} \sum_{j=1}^{J} s(\boldsymbol{x}_j). \tag{33}$$

to evaluate how good the clustering solution is. Higher $\tilde{s}$'s are associated with a better clustering solution.

In the study of Vendramin et al. (2010), the silhouette index had the most robust performance with regard to the different evaluation scenarios. The other evaluation criteria were more sensitive to the dimensionality of the data and the true number of clusters.

Appendix D. Empirical distribution of the category-specific weighted average damage rates and expected claim frequencies
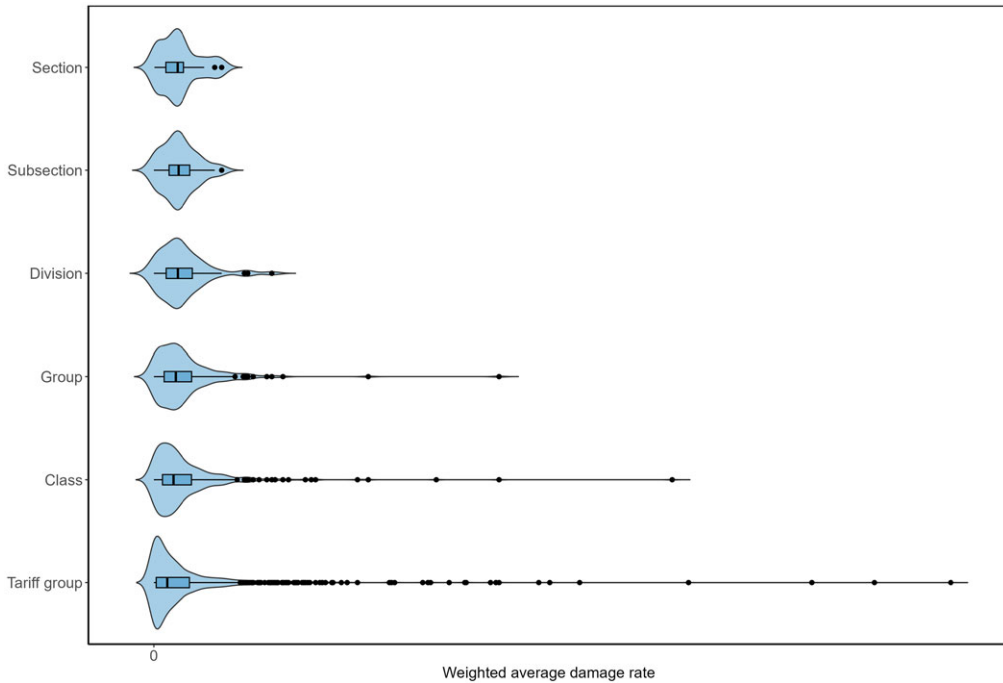
Figure D.1. Empirical distribution of the category-specific weighted average damage rate at different levels in the hierarchy. One large value is removed to obtain a better visualization.
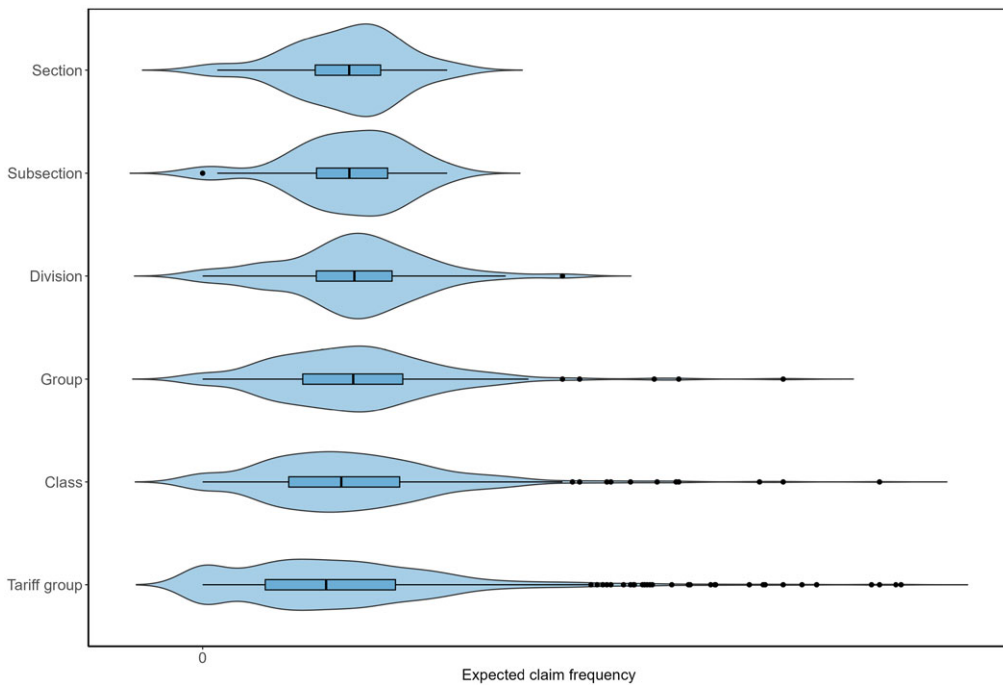

Figure D.2. Empirical distribution of the category-specific expected claim frequency at different levels in the hierarchy. Two large values are removed to obtain a better visualization.

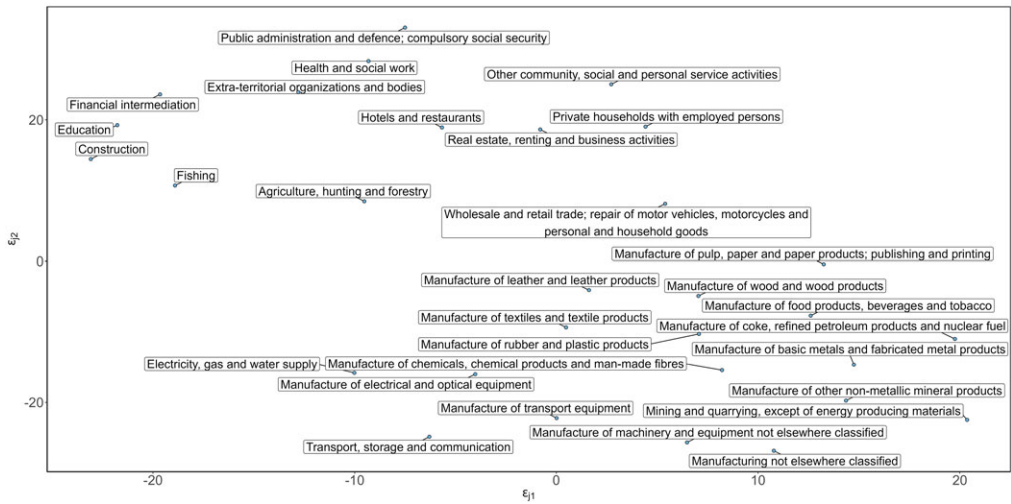## Appendix E. Low-dimensional representation of the embedding vectors

Figure E.1. Low-dimensional visualization of all embedding vectors, resulting from the pre-trained USE v4 encoder, constructed for different categories at the `subsection` level. The text boxes display the textual labels. The blue dots connected to the boxes depict the position in the low-dimensional representation of the embeddings.

Figure E.2. Low-dimensional visualization of all embedding vectors, resulting from the pre-trained USE v5 encoder, constructed for different categories at the `subsection` level. The text boxes display the textual labels. The blue dots connected to the boxes depict the position in the low-dimensional representation of the embeddings.
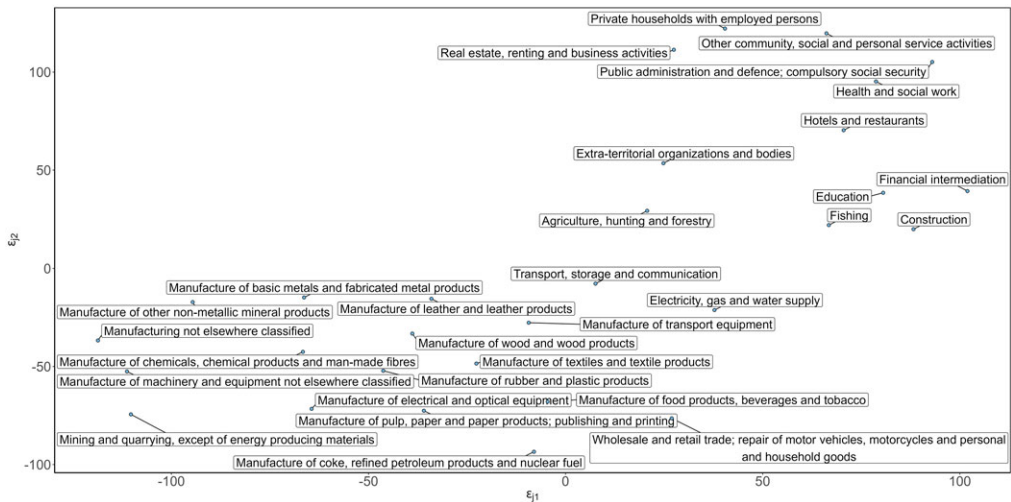
Appendix F. Predictive performance when using the angular distance matrix $\mathcal{D}$ for the cluster evaluation criteria

Table F.1.   Predictive performance on the training and test set, when the internal evaluation criterion is calculated using the angular distance and the complete feature vector

	J'	$\sum_{j'=1}^{J'} K'_{j'}$	Development Gini index	Validation Gini index	Loss ratio
Benchmark	18	641	0.658	0.585	1.006
HCA:					
Silhouette index	7	137	0.669	0.625	1.008
Dunn index	14	322	0.662	0.604	1.008
Davies-Bouldin index	14	316	0.670	0.613	1.007
CH index	13	157	0.658	0.604	1.011
k-medoids:					
Silhouette index	6	135	0.655	0.593	1.008
Dunn index	14	315	0.656	0.576	1.006
Davies-Bouldin index	14	303	0.674	0.621	1.011
CH index	10	125	0.658	0.627	1.010
Spectral clustering:					
Silhouette index	8	150	0.670	0.599	1.009
Dunn index	14	299	0.663	0.585	1.008
Davies-Bouldin index	16	323	0.668	0.583	1.006
CH index	16	224	0.671	0.619	1.008