A self-organizing predictive map for non-life insurance

Donatien Hainaut*
ISBA, Université Catholique de Louvain, Belgium

December 9, 2018

Abstract

This article explores the capacity of self-organizing maps (SOMs) for analysing non-life insurance data. Contrary to feed forward neural networks, also called perceptron, a SOM does not need any a priori information on the relevancy of variables. During the learning procedure, the SOM algorithm selects the most relevant combination of explanatory variables and reduces by this way the collinearity bias. However, the specific features of insurance data require adapting the classic SOM framework to manage categorical variables and the low frequency of claims. This work proposes several extensions of SOMs in order to study the claims frequency of a portfolio of motorcycle insurances. Results are next compared to these computed with variants of the k-mean clustering algorithm.

KEYWORDS : Neural network, self organizing map, non-life insurance, claims frequency, regression models, k-means algorithm

1 Introduction

Among neural models, the Multilayer Perceptron (see e.g. Hertz et al. (1991)) is one of the most popular. This model is an extension of the Rosenblatt's perceptron (1958) and is inspired by the functioning of neural cells constituting animal brains. This model admits both quantitative and categorical variables as inputs and is efficient to perform classification or non linear regression analysis. Prior to year 2010, neural networks were mainly applied to evaluate insurer's credit risk or fraud detection. E.g. Brockett et al. (1994) and (2006) train a multilayer perceptron to detect insurer insolvency. Whereas Viane et al. (2002) compare classification techniques for detection of expert automobile insurance fraud.

The use of neural networks in actuarial sciences is relatively recent. Ogunnaike and Si (2017) use a feed-forward neural network to predict insurance claim severity losses. Sakthivel and Rajitha (2017) compare the performance of Poisson regression models with

*Postal address: Voie du Roman Pays 20, 1348 Louvain-la-Neuve (Belgium) . E-mail to: donatien.hainaut(at)uclouvain.be

neural networks to forecast the claims number. Wüthrich and Buser (2017) provide an excellent overview of the usefulness of these methods for actuaries. In life insurance, Hainaut (2017) proposes a perceptron-based model to forecast future mortality.

The perceptron is a method with supervised learning. It means that we have to a priori identify the most relevant variables and to know the desired outputs for combinations of these variables. For example, forecasting the frequency of car accidents with a perceptron requires an a priori segmentation of some explanatory variables like the driver's age into categories, in a similar manner to Generalized Linear Models (see e.g. Ohlsson and Johansson (2010)). The misspecification of these categories can induce a large bias in the forecast. On the other hand, the presence of collinearity between covariates affects the accuracy of the prediction. In this situation, the coefficient estimates of the multiple regression may change erratically in response to small changes in the model or the data. Self-organizing maps offer an elegant solution to segment explanatory variables and to detect dependence among covariates.

Self-organizing maps are artificial neural networks that do not ask any a priori information on the relevancy of variables. For this reason, they belong to the family of unsupervised algorithms. This method developed by Kohonen (1982) aims to analyze the original information by simplifying the amount of rough data, computing some basic features, and giving visual representation. More precisely, SOMs produce a low dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to perform reduction of dimensions. As perceptrons, SOMs are used for fraud detection (Brockett et al., 1998) or failure prediction (Huysmans et al. 2006).

This article contributes to the literature both from an empirical and methodological points of view. At the best of our knowledge, self organizing maps have not been applied yet to study non-life insurance data. The features of insurance data are: a low frequency of claims, a high number of policyholders and a mix of quantitative and categorical variables. These distinctive characteristics require to adapt the Kohonen's algorithm in several directions. First, we propose in section 2 a Bayesian extension of SOMs in order to regress the claims frequency on quantitative explanatory variables. Without recourse to a Bayesian assumption, it is not possible to obtain a reliable estimate of claims frequencies. Mainly because in this case the empirical frequency is null for many sub-groups of policyholders due to the low number of claims. We compare results to these obtained with a Bayesian version of the k-means algorithm. Second, the original Kohonen's algorithm was designed to study quantitative variables. In sections 3 and 4, we propose a SOM to study categorical variables. In a similar manner to multiple correspondence analysis, we define a χ^2 distance between combinations of categories. This approach, based on the article of Cottrel et al. (2004) allows us to regroup categorical variables and to detect dominant features in the portfolio of contracts. Finally, the theory introduced in later sections serves us to develop

a SOM and a k-means algorithm for regressing claims frequency on quantitative and categorical variables. Empirical results reveal that this type of SOM is a powerful tool for segmenting policyholders.

2 The self-organizing maps (SOM)

Self-organizing maps offer an elegant solution to segment explanatory variables and to detect dependence among covariates. Initially, Kohonen (1982) developed SOMs for analysing quantitative variables. In the first subsection, we present this algorithm and apply it to an insurance data set to regroup policyholders with respect to quantitative variables. In section 2.2, we show how SOMs may be adapted in order to regress the claims frequency on quantitative variables in a Bayesian framework.

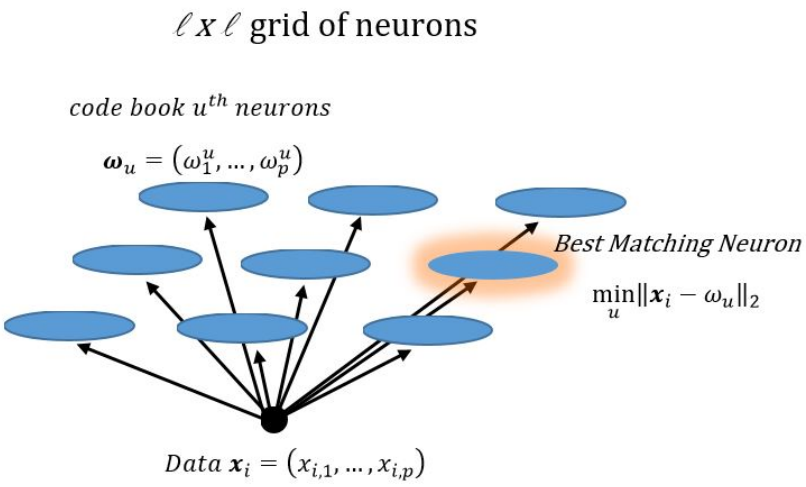

Figure 1: Illustration of the grid of neurons. The data is sent to all neurons and the best matching neuron is the only one to be activated. The best matching neuron has the closest codebook to data among all neurons.

2.1 The SOM for quantitative variables

In this section, features of insurance policies are exclusively quantitative variables. The number of insurance policies is denoted by n. Each of these policies is described by p real-valued variables. The data are arranged in a table X with n rows and p columns. The rows of table X are denoted by $\boldsymbol{x}_i = X_{i,\cdot}$ and are the inputs of the SOM algorithm.

As illustrated in figure 1, the map is represented by a two-dimensional grid, with l by l nodes or neurons, indexed by $u = 1, ..., l^2$. This grid is described by a $l^2 \times 2$ matrix C. C contains the coordinates of nodes in $\mathbb{R}^2$. In this article, the domain on which is defined the grid is the unit square $[0, 1] \times [0, 1]$. Neurons are equally spaced on this pavement.

In order to define the neighbourhood of a neuron in this grid, we introduce a topological distance between neurons. Here, we use the Euclidian distance between lines u and v of the matrix C: $\|C_{u,.} - C_{v,.}\|_2$.

Algorithm 1 Kohonen's algorithm for quantitative variables.

Initialization :

Randomly attribute codebooks $\boldsymbol{\omega}_1(0)$,..., $\boldsymbol{\omega}_{l^2}(0)$.

Main procedure :

For $e = 0$ to maximum epoch, e_{max}

For $i = 1$ to n

1) Find the node u matching at best the features of the i^{th} policy

$$u \quad = BMN(i) = \quad \arg\min_u d(\boldsymbol{x}_i, \boldsymbol{\omega}_u(e)) \,. \tag{1}$$

This node is the BMN (best matching node).

2) Update codebooks in the neighbourhood of the BMN by pulling them closer in $\mathbb{R}^p$ to the input vector:

$$\boldsymbol{\omega}_v(e+1) = \boldsymbol{\omega}_v(e) + \theta(u, v, e)\, \epsilon(e)\, (\boldsymbol{x}_i - \boldsymbol{\omega}_v(e)) \tag{2}$$

for $v = 1, ..., l^2$ and where

$$\epsilon(e) \quad = \quad \epsilon_0 \left(\frac{e_{max} - e}{e_{max}} \right) , \tag{3}$$

$$\theta(u, v, e) \quad = \quad \theta_0 \exp\left(- \frac{\left(\|C_{u,.} - C_{v,;}\|_2 \right)^2}{2\sigma(e)^2} \right) , \tag{4}$$

$$\sigma(e) \quad = \quad \sigma_0 \left(\frac{1.2\, e_{max} - e}{e_{max}} \right) . \tag{5}$$

End loop on policies, i

3) Calculation of the total distance d^{total} between policies and BMNs:

$$d^{total} = \sum_{i=1}^{n} \left\| \boldsymbol{\omega}_{BMN(i)} - \boldsymbol{x}_i \right\|_2 .$$

End loop on epochs, e

We associate to each node a codebook denoted by $\boldsymbol{\omega}_u = \{\omega_1^u, ..., \omega_p^u\}$ that is an $\mathbb{R}^p$ vector of weights for $u = 1, ..., l^2$. We pursue a double objective. First, we wish to associate a node of the grid to each insurance policy and partition the portfolio in l^2 clusters. Second, we want to determine codebook vectors $\boldsymbol{\omega}_u$ for $u = 1, ..., l^2$ containing the average profile

of policies assigned to node u. The vector $\boldsymbol{\omega}_u$ may be seen as the center of gravity in $\mathbb{R}^d$ of policies associated to the u^{th} neuron. The definition of this barycenter requires the definition of a distance between the u^{th} node and the i^{th} insurance policy, whose respective positions in $\mathbb{R}^p$ are identified by $\boldsymbol{\omega}_u$ and $\boldsymbol{x}_i$. In the case of quantitative variables, we use the Euclidian distance:

$$\begin{aligned} d(\boldsymbol{x}_i, \boldsymbol{\omega}_u) &= \|\boldsymbol{\omega}_u - \boldsymbol{x}_i\|_2 \qquad u = 1, .., l^2 \; i = 1, ..., n \\ &= \sqrt{\sum_{k=1}^{p} \left(\omega_k^u - x_{i,k}\right)^2}. \end{aligned}$$

Remark that the neural map admits a double representation. One is on a pavement $[0,1] \times [0,1]$ and positions of neurons are fixed. The other one is in $\mathbb{R}^d$ where the positions of nodes are determined by the p-vectors $\boldsymbol{\omega}_u$ for $u = 1, ..., l^2$. Kohonen (1982) proposed a procedure to construct this map that is recalled in Algorithm 1. The algorithm scans the portfolio and finds for each policy the neuron with the closest codebook to its features. This neuron is called the best matching node (BMN). After this step, weights of neurons in the neighbourhood of this BMN are updated in the direction of the policy features. The size of the update is proportional to the epoch of the algorithm and to the distance between the BMN and updated neurons. Notice that functions $\epsilon(e)$ and $\sigma(e)$ in equations (3) and (5) may be replaced by any other decreasing functions of the epoch, e. The total distance d^{total} is the error of classification if we use the feature $\boldsymbol{\omega}_{BMN(i)}$ for the i^{th} policy, instead of the real one $\boldsymbol{x}_i$. This distance is monitored to check the convergence of the algorithm. When it does not vary anymore, the learning of the neural net is finished.

The speed of convergence of the Kohonen's algorithm depends on initial weights $\boldsymbol{\omega}_u(0)$. They should be chosen in order to reflect as much as possible the largest set of features of policies[1]. The convergence also depends on parameters, ϵ_0, θ_0 and σ_0 of equations (3), (4) and (5). If they are too high, weights of neurons oscillate during the first iterations. If ϵ_0, θ_0 and σ_0 are too small, modifications of the codebook are not enough significant. In both case, the number of epochs must be increased to achieve convergence.

To illustrate this section, we apply the Kohonen algorithm to data from the Swedish insurance company *Wasa*, before its fusion with *Länsförsäkringar Alliance* in 1999. The data set is available on the companion website of the book of Ohlsson and Johansson (2010) and contains information about motorcycles insurances over the period 1994-1998. Each policy is described by quantitative and categorical variables. The quantitative variables are the insured's age and the age of his vehicle. The categorical variables are: the policyholder's gender, the geographic zone and the category of the vehicle. The category of the vehicle is based on the ratio power in KW $\times 100$ / vehicle weight in kg + 75, rounded to the nearest integer. The database also reports the number of claims, the total claim costs and the

[1] In our approach, codebooks are randomly chosen. An alternative consists to use the initialization procedure of the k-means algorithm, presented in Section 2.2.

duration of the contract for each policies. Table 1 summarizes the information provided by categorical variables.

Rating factors	**Class**	**Class description**
Gender	M	Male
	K	Female
Geographic area	1	Central and semi-central parts of Sweden's three largest cities
	2	Suburbs plus middle-sized cities
	3	Lesser towns, except those in 5 or 7
	4	small towns and countryside
	5	Northen towns
	6	Northen countryside
	7	Gotland (Sweden's largest island)
Vehicle class	1	EV ratio -5
	2	EV ratio 6-8
	3	EV ratio 9-12
	4	EV ratio 13-15
	5	EV ratio 16-19
	6	EV ratio 20-24
	7	EV ratio 25-

Table 1: Rating factors of motorcycle insurances. Source: Ohlsson and Johansson (2010).

The database counts n =62436 policies after removing contracts with a null duration. In this section, we focus on two quantitative variables: the owner's age and age of the vehicle. Before running the Kohonen's algorithm we scale the variables in order to convert their domain on the pavement $[0,1] \times [0,1]$. The columns of the matrix X respectively contain the rescaled ages of owners and vehicles as follows

$$x_{i,1} = \frac{Age\, i^{th}\, policyholder}{Max(Age)} \quad , \quad x_{i,2} = \frac{Vehicle\, Age\, i^{th}\, policy}{Max(Vehicle\, Age)}\,. \tag{6}$$

We build a map of the portfolio that regroups policyholders according to these two criteria. The number of epochs is $e_{max} = 100$ and the grid of neurons count 9 elements ($l = 9$). The initial codebooks are chosen in order to regularly cover the $[0,1] \times [0,1]$ pavement. The parameters of equations (3), (4) and (5) for the update of codebooks are: $\epsilon_0 = 0.01$, $\theta_0 = 1$ and $\sigma_0 = 0.10$. These values have been chosen by trials and errors in order to ensure a quick convergence.

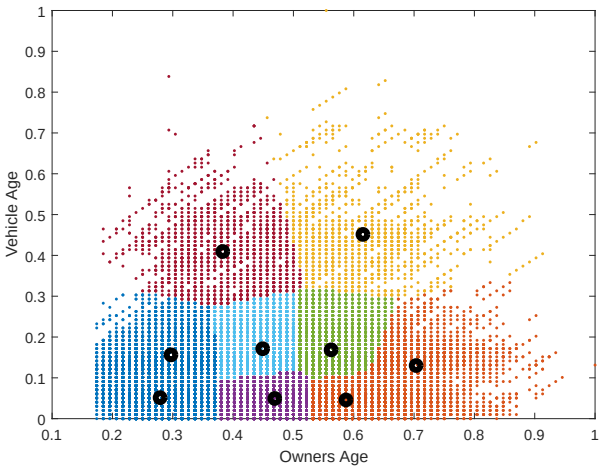

Figure 2: Kohonen's map of the portfolio, with 9 neurons. The segmenting variables are the owner's and vehicle ages. Each policy is represented by a dot. Black dots point out the position of neurons. The colors identify the area of influence of neurons.

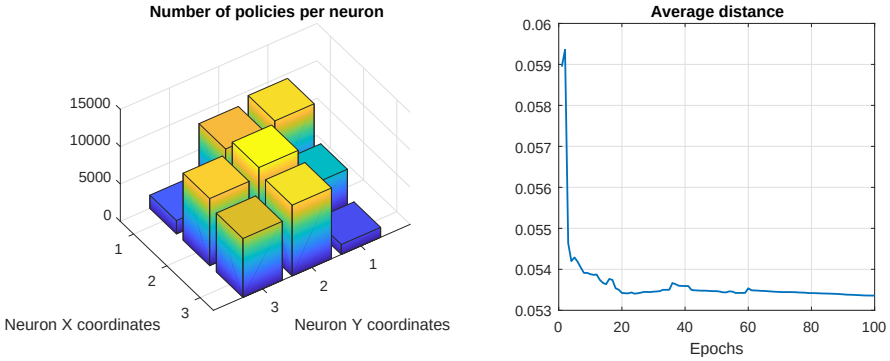

Figure 3: The left graph shows the number of policies assigned to each neuron. The right graph plots the evolution of the average distance $\frac{d^{total}}{n}$ over iterations.

Figure 2 shows the Kohonen's map in the space of variables in which policies are identified by a dot. Each colored area represents a group of policies centered around a neuron (black dots). The right graph of Figure 3 reports the evolution of the distance $\frac{d^{total}}{n}$ between policies and BMN codebooks. This distance is nearly stable and the convergence is achieved after 60 iterations. The right graph of this figure shows the number of policies associated to each neurons. Two neurons are coupled to less than 3% of policies. The others are associated to clusters of policies representative of 7% to 17% of the total portfolio. Table 2 reports the codebooks of neurons. $\omega_1^u \times max(Age)$ and $\omega_1^u \times max(Veh.Age)$ for $u = 1$ to 9 are respectively the average owner's and vehicle ages of the u^{th} subset of policies. An analysis of the average frequency of claims per cluster reveals that this frequency tends

to decrease with the owner's age. However, the pooling of policies is based only on ages of policyholders and vehicles, not on claims frequency. In the next section, we modify the Kohonen's algorithm in order to delimit subsets of policies with homogeneous claims frequencies.

Neuron u	$\omega_1^u \times max(Age)$	$\omega_2^u \times max(Veh.Age)$	$\bar{\lambda}_u$	% of the population
1	25.77	4.96	0.0413	14.96
2	27.32	15.34	0.0154	14.39
3	35.27	40.41	0.0052	2.86
4	41.36	16.85	0.0038	15.21
5	43.22	4.77	0.0096	13.59
6	51.81	16.58	0.0034	16.43
7	54.06	4.43	0.0086	12.63
8	56.66	44.60	0	2.14
9	64.78	12.77	0.0058	7.79

Table 2: This table reports the codebooks of neurons, the average frequency of claims for clusters of policies associated to each neuron, and cluster relative sizes.

2.2 Comparison of SOM and k-means clustering

Self-organizing maps produce similar results to the method of k-means clustering. The codebook of a neuron in a SOM contains the coordinates of the center of gravity of a cluster of policies. In the method of k-means, the codebook of a neuron is called a centroid but must be interpreted in the same way. The main difference lies in the procedure for estimating these codebooks. The purpose of SOM and k-means methods is to partition a cloud of n points in $\mathbb{R}^p$ into $k = l \times l$ clusters. In this section, we briefly remind the k-means algorithm. In this approach, the coordinates of the u^{th} centroid is contained in a vector $\boldsymbol{c}_u = (c_1^u, \ldots, c_p^u)$ for $u = 1, ..., k$. For a given distance $d(.,.)$ and a set of k centroids, we define the clusters or classes of data S_u for $u = 1, ..., k$ as follows:

$$S_u = \{\boldsymbol{x}_i \ : \ d(\boldsymbol{x}_i, \boldsymbol{c}_u) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_j) \ \forall j \in \{1, ..., k\}\} \quad u = 1, ..., k\,. \tag{7}$$

Here, $d(.,.)$ is the Euclidian distance but other distances can be considered. The center of gravity of S_u is a p vector $\boldsymbol{g}_u = \left(g_1^u, \ldots, g_p^u\right)$ such that

$$\boldsymbol{g}_u = \frac{1}{|S_u|} \sum_{\boldsymbol{x}_i \in S_u} \boldsymbol{x}_i\,.$$

The center of gravity of the full dataset is denoted by $\boldsymbol{g} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i$. We define the global inertia by

$$I_X = \frac{1}{n} \sum_{i=1}^{n} d\left(\boldsymbol{x}_i, \boldsymbol{g}\right)^2\,,$$

and the inertia I_u of a cluster S_u by

$$I_u = \sum_{\boldsymbol{x}_i \in S_u} \frac{1}{|S_u|} d\left(\boldsymbol{x}_i, \boldsymbol{g}_u\right)^2 \quad u = 1, ..., k.$$

The interclass inertia I_c is the inertia of the cloud of centers of gravity:

$$I_c = \sum_{u=1}^{k} \frac{|S_u|}{n} d\left(\boldsymbol{g}_u, \boldsymbol{g}\right)^2 ,$$

whereas the intraclass inertia I_a is the sum of clusters inertiae, weighted by their size:

$$\begin{aligned} I_a &= \sum_{u=1}^{k} \frac{|S_u|}{n} I_u \\ &= \frac{1}{n} \sum_{u=1}^{k} \sum_{\boldsymbol{x}_i \in S_u} d\left(\boldsymbol{x}_i, \boldsymbol{g}_u\right)^2 . \end{aligned}$$

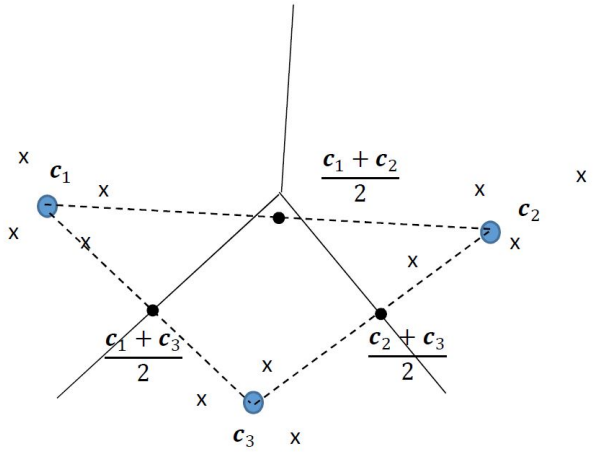

Figure 4: Illustration of the partition of a dataset with the k-means algorithm.

According to the König-Huyghens theorem, the total inertia is the sum of the intraclass and interclass inertiae: $I_X = I_c + I_a$. An usual criterion of classification consists to seek for a partition of X minimizing the intraclass inertia I_a in order to have homogeneous clusters on average. This is equivalent to determine the partition maximizing the interclass inertia, I_c.

Algorithm 2 Lloyd's algorithm for k-means clustering.

Initialization:

Randomly set up initial positions of centroids $\boldsymbol{c}_1(0)$,...,$\boldsymbol{c}_k(0)$.

Main procedure:

For $e = 0$ to maximum epoch, e_{max}

Assignment step:

For $i = 1$ to n

1) Assign $\boldsymbol{x}_i$ to a cluster $S_u(e)$ where $u \in \{1, ..., k\}$

$$S_u(e) = \{\boldsymbol{x}_i \, : \, d(\boldsymbol{x}_i, \boldsymbol{c}_u(e)) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_j(e)) \; \forall j \in \{1, ..., k\}\} \, .$$

End loop on policies, i.

Update step:

For $u = 1$ to k

2) Calculate the new centroids $\boldsymbol{c}_u(e+1)$ of $S_u(e)$ as follows

$$\boldsymbol{c}_u(e+1) = \frac{1}{|S_u(e)|} \sum_{\boldsymbol{x}_i \in S_u(e)} \boldsymbol{x}_i \, .$$

End loop on centroids, u.

3) Calculation of the total distance d^{total} between observations and closest centroids:

$$d^{total} = \sum_{u=1}^{k} \sum_{\boldsymbol{x}_i \in S_u(e)} d(\boldsymbol{x}_i, \boldsymbol{c}_u(e+1)) \, .$$

End loop on epochs e

This problem is computationally difficult (NP-hard). However, there exist efficient heuristic procedures converging quickly to a local optimum. The most common method uses an iterative refinement technique called the k-means or Lloyd's method (1957) which is detailed in Algorithm 2. Given an initial set of k random centroids $\boldsymbol{c}_1(0)$,...,$\boldsymbol{c}_k(0)$, we construct a partition $\{S_1(0), \ldots, S_k(0)\}$ of the dataset according to the rule in equation (7). This partition is a set of convex polyhedrons delimited by median hyperplans of centroids as illustrated in Figure 4. Next, we replace the k random centroids by the k centers of gravity $(\boldsymbol{c}_u(1))_{u=1:k} = (\boldsymbol{g}_u(0))_{u=1:k}$ of these classes and we iterate till convergence. At each iteration, we can prove that the intraclass inertia is reduced.

The Lloyd's algorithm proceeds by alternating between two steps. In the assignment step of the e^{th} iteration, we associate each observation $\boldsymbol{x}_i$ to a cluster $S_u(e)$ whose centroid $\boldsymbol{c}_u(e)$ has the least distance, $d(\boldsymbol{x}_i, \boldsymbol{c}_u(e))$. This is intuitively the nearest centroid to each observation. In the update step, we calculate the new means $\boldsymbol{g}_u(e)$ to be the centroids

$\boldsymbol{c}_u(e+1)$ of observations in new clusters[2]. The algorithm converges when the assignments no longer change. There is no guarantee that a global optimum is found using this algorithm. The k-means++ algorithm of Arthur and Vassilvitskii (2007) uses an heuristic to find centroid seeds for k-means clustering. The procedure to initialize the k-means heuristic is detailed in Algorithm 3. It improves the running time of the Lloyd's algorithm, and the quality of the final solution. It can also be used for initializing codebooks of SOM algorithms presented in this article.

Algorithm 3 Initialization of centroids for the k-means algorithm.

Initialization :

Select an observation uniformly at random from the data set, X. The chosen observation is the first centroid, and is denoted $\boldsymbol{c}_1(0)$.

Main procedure:

For $j = 2$ to k

For $i = 1$ to n

1) Calculate the distance $d(\boldsymbol{x}_i, \boldsymbol{c}_{j-1}(0))$ from $\boldsymbol{x}_i$ to $\boldsymbol{c}_{j-1}(0)$.

End loop on policies, i

2) Select the next centroid, $\boldsymbol{c}_j(0)$ at random from X with probability

$$\frac{d^2(\boldsymbol{x}_i, \boldsymbol{c}_{j-1}(0))}{\sum_{i=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{c}_{j-1}(0))} \quad i = 1, ..., n.$$

End loop on k

Neuron u	$c_1^u \times max(Age)$	$c_2^u \times max(Veh.Age)$	$\bar{\lambda}_u$	% of the population
1	25.53	5.29	0.0404	15.70
2	28.18	15.65	0.0140	14.71
3	35.31	40.19	0.0052	2.91
4	43.01	16.65	0.0040	17.87
5	43.67	4.55	0.0092	14.47
6	53.49	16.47	0.0033	14.38
7	54.75	4.42	0.0091	11.51
8	56.62	44.63	0	2.14
9	65.97	12.44	0.0055	6.31

Table 3: This table reports the positions of centroids, the average frequency of claims for clusters of policies associated to each centroids, and cluster relative sizes.

[2]A variant of this algorithm consists to recompute immediately the new position of centroids after assignment of each records of the dataset.

Table 3 reports the results of the Lloyd's algorithm applied to the dataset X of rescaled owner's and vehicle ages. A comparison with figures of Table 2 reveals the similarity of neural codebooks and centroids positions. However, in terms of total distance, the SOM slightly outperforms the k-means algorithm. After 100 iterations, we obtain a total distance of d^{total} =3331.5 for the SOM whereas the total distance for the k-means algorithm is equal to d^{total} =3338.5.

2.3 A Bayesian regressive SOM with quantitative variables

In this section, we aim to assign policies to groups in which we observe relatively homogeneous claims frequencies. The SOM plays then two roles. Firstly, it segments policies based on their features. Secondly, it explains the claims frequency as a function of these characteristics. For this reason, the SOM is said regressive. For the moment, policies are exclusively described by quantitative variables stored in a table X with n rows and p columns. The the number of claims is denoted by $(N_i)_{i=1,...n}$ and durations of the contract are $(\nu_i)_{i=1,...n}$. By nature, the number of claims N_i is null for most of policies (693 claims for 62 436 policies). The neural map is a $l \times l$ array of neurons with a matrix C that contains the coordinates of nodes in $\mathbb{R}^2$. The codebook of the u^{th} neurons is still denoted by $\boldsymbol{\omega}_u = \{\omega_1^u, ..., \omega_p^u\}$.

We wish to construct a map of the portfolio in which policies are bundled into groups identified by a neuron. We denote by $\Omega_{u=1,...,l^2}$ the cluster of insurance contracts assigned to the u^{th} neuron. We assume that the number of claims caused by a policy in Ω_u is distributed according to a Poisson law of parameter λ_u. If the total exposure is $\nu^{\Omega_u} = \sum_{i \in \Omega^u} \nu_i$, the distribution of the total number of claims in Ω_u, noted N^{Ω_u} is given by

$$P\left(N^{\Omega_u} = m\right) = \frac{\left(\lambda_u \nu^{\Omega_u}\right)^m}{m!} e^{-\lambda_u \nu^{\Omega_u}}.$$

The log-likelihood estimator of λ_u is then equal to:

$$\bar{\lambda}_u = \frac{N^{\Omega_u}}{\nu^{\Omega_u}} = \frac{\sum_{i \in \Omega^u} N_i}{\sum_{i \in \Omega^u} \nu_i}.$$

The homogeneity of claims frequencies inside a subset Ω_u is measured by the realized standard deviation of N^{Ω_u}:

$$d_{\Omega_u}(\bar{\lambda}_u) = \sqrt{\sum_{i \in \Omega_u} \left(N_i - \bar{\lambda}_u \nu_i\right)^2},$$

and the distance between the realized frequency of the i^{th} policy and average frequency for the subgroup Ω_u is given by

$$d_{\Omega_u}(i, \bar{\lambda}_u) = \sqrt{\left(N_i - \bar{\lambda}_u \nu_i\right)^2}.$$

A first idea could be using this standard deviation to define a new distance between the i^{th} policy and the u^{th}, as follows:

$$\begin{aligned} d(\boldsymbol{x}_i, \boldsymbol{\omega}_u, \bar{\lambda}_u) &:= \|\boldsymbol{\omega}_u - \boldsymbol{x}_i\|_2 + \beta \, d_{\Omega_u}(i, \bar{\lambda}_u) \qquad u = 1, ..., l^2 \; i = 1, ..., n \\ &= \sqrt{\sum_{k=1}^{p} \left(\omega_k^u - x_{i,k}\right)^2} + \beta \sqrt{\left(N_i - \bar{\lambda}_u \nu_i\right)^2} \end{aligned} \tag{8}$$

where β is a weight adjusting the regressive feature of the map with respect to its segmentation function. This distance could be used in equation (1) of the first step of Kohonen's algorithm. Whereas $\bar{\lambda}_u$ would be updated at the end of each iteration, in the third step of Algorithm 1. However, this approach is not satisfactory when applied to the insurance data. Given that the database counts only 693 claims for 62 436 policies, the algorithm tends to discriminate the portfolio into two subsets: one regrouping policies with no claim and the other one gathering policies with one or more claims. To remedy to this issue, we propose a Bayesian approach.

We consider a Gamma random variable $\Theta \sim \Gamma(\gamma, \gamma)$ and assume that conditionally to Θ, the r.v. $N^{\Omega_1}, ..., N^{\Omega_N}$ are independent with conditional distributions:

$$N_{u|\Theta} \sim Poi\left(\lambda_u \Theta \nu^{\Omega_u}\right) \quad u = 1, ..., l^2 \tag{9}$$

where $\lambda_u > 0$ and γ are the prior frequency of claims and a dispersion parameter. The choice of λ_u and γ is discussed at the end of this section. Under the Bayesian assumption, the coefficient of variation of the expected frequency is then $\gamma \frac{1}{\gamma^2} = \frac{1}{\gamma}$. We have the following standard result:

Proposition 2.1. *The posterior expected frequency on Ω_u given observations N^{Ω_u} is given by*

$$\mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) = \alpha_u \bar{\lambda}_u + (1 - \alpha_u) \, \lambda_u \,, \tag{10}$$

where $\bar{\lambda}_u = \frac{N^{\Omega_u}}{\nu^{\Omega_u}} = \frac{\sum_{i \in \Omega^u} N_i}{\sum_{i \in \Omega^u} \nu_i}$ and with credibility weights

$$\alpha_u = \frac{\nu^{\Omega_u} \lambda_u}{\gamma + \nu^{\Omega_u} \lambda_u} \in (0, 1) \,. \tag{11}$$

for $u = 1, ..., l^2$.

Proof: Using the Bayes rule, the probability density function of $\Theta|N_{\Omega^u}$ is rewritten as

$$\begin{aligned} f_\Theta(\vartheta \,|\, N_{\Omega^u} = n) &= \frac{f_\Theta(\vartheta) P(N_{\Omega^u} = n|\Theta = \vartheta)}{P(N_{\Omega^u} = n)} \,. \\ &= \frac{\frac{\gamma^\gamma}{\Gamma(\gamma)} \vartheta^{\gamma-1} e^{-\gamma\vartheta} e^{-\lambda_u \vartheta \nu^{\Omega_u}} \frac{\left(\lambda_u \vartheta \nu^{\Omega_u}\right)^n}{n!}}{\int \frac{\gamma^\gamma}{\Gamma(\gamma)} \vartheta^{\gamma-1} e^{-\gamma\vartheta} e^{-\lambda_u \vartheta \nu^{\Omega_u}} \frac{(\lambda_u \vartheta \nu^{\Omega_u})^n}{n!} d\vartheta} \\ &\propto \vartheta^{\gamma+n} e^{-(\gamma+\lambda_u \nu^{\Omega_u})\vartheta} \end{aligned}$$

where $\propto$ is the proportional operator. $f_\Theta(\vartheta \,|\, N_{\Omega^u})$ is the density of a Gamma distribution with the updated parameters:

$$\begin{aligned} \gamma' &= \gamma + N_{\Omega^u} \,, \\ \beta' &= \gamma + \lambda_u \nu^{\Omega_u} \,. \end{aligned}$$

As $\bar{\lambda}_u = \frac{N^{\Omega_u}}{\nu^{\Omega_u}} = \frac{\sum_{i\in\Omega^u} N_i}{\sum_{i\in\Omega^u} \nu_i}$ then

$$\begin{aligned} \mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) &= \lambda_u \mathbb{E}\left(\Theta | N^{\Omega_u}\right) \\ &= \lambda_u \frac{\gamma + N^{\Omega_u}}{\gamma + \lambda_u \nu^{\Omega_u}} \\ &= \lambda_u \frac{\gamma}{\gamma + \lambda_u \nu^{\Omega_u}} + \frac{\lambda_u \nu^{\Omega_u}}{\gamma + \lambda_u \nu^{\Omega_u}} \frac{N^{\Omega_u}}{\nu^{\Omega_u}} \\ &= (1 - \alpha_u)\, \lambda_u + \alpha_u \bar{\lambda}_u \,. \end{aligned}$$

■

For a given prior frequency λ_u and dispersion parameter γ, the posterior mean $\mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right)$, is an estimator of the expected frequency on Ω_u. Contrary to the empirical intensity $\bar{\lambda}_u$, this posterior is always strictly positive. However in practice, the prior λ_u is unknown. For this reason, we opt for an empirical version of the credibility estimator (9):

$$\mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) = \alpha_u \bar{\lambda}_u + (1 - \alpha_u)\, \bar{\lambda} \,, \tag{12}$$

where $\bar{\lambda}_u = \frac{N^{\Omega_u}}{\nu^{\Omega_u}}$ and $\bar{\lambda} = \frac{\sum_{u=1}^{l^2} N^{\Omega_u}}{\sum_{u=1}^{l^2} \nu^{\Omega_u}}$ is the global average claims frequency. Whereas we choose credibility weights equal to

$$\alpha_u = \frac{\nu^{\Omega_u} \bar{\lambda}}{\gamma + \nu^{\Omega_u} \bar{\lambda}} \,.$$

A similar assumption is done in the R package "rpart" for Poisson regression trees. In the Bayesian framework, the homogeneity of claims frequency inside a subset Ω_u is measured by the realized standard deviation of N^{Ω_u}, calculated with the credibility estimator:

$$d_{\Omega_u}(\bar{\lambda}_u) = \sqrt{\sum_{i\in\Omega_u} \left(N_i - \mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) \nu_i\right)^2} \,,$$

and the distance between the realized frequency of the i^{th} policy and the posterior expected frequency for the subgroup Ω_u is given by

$$d_{\Omega_u}(i, \bar{\lambda}_u) \quad = \quad \sqrt{\left(N_i - \mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) \nu_i\right)^2},$$

whereas the distance between the i^{th} policy and the u^{th} neuron in the Kohonen's algorithm becomes:

$$\begin{aligned} d(\boldsymbol{x}_i, \boldsymbol{\omega}_u, \bar{\lambda}_u) &:= \|\boldsymbol{\omega}_u - \boldsymbol{x}_i\|_2 + \beta \, d_{\Omega_u}(i, \bar{\lambda}_u) \qquad u = 1, ..., l^2 \; i = 1, ..., n \\ &= \sqrt{\sum_{k=1}^{p} \left(\omega_k^u - x_{i,k}\right)^2} + \beta \sqrt{\left(N_i - \mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) \nu_i\right)^2} \end{aligned} \tag{13}$$

where β is a weight adjusting the regressive feature of the map with respect to its segmentation capacity. Algorithm 4 summarizes the steps of the Bayesian regressive SOM. We run the Algorithm 4 with 9 neurons. The weight β in the definition of the distance (13) is set to 10. The credibility factor γ is fixed to 4. The convergence is achieved in less than 100 iterations.

Neuron u	$\omega_1^u \times$ $max(Age)$	$\omega_2^u \times$ $max(Veh.\,Age)$	$\mathbb{E}\left(\lambda_u \Theta \| N^{\Omega_u}\right)$	% of the population
1	26.64	4.41	0.0019	13.89
2	26.96	15.27	0.0005	15.97
3	34.51	8.02	0.7178	0.91
4	40.50	6.53	0.1204	0.26
5	41.70	16.63	0.0003	18.13
6	45.58	44.35	0.0016	4.15
7	47.59	4.43	0.0002	20.56
8	53.27	16.68	0.003	16.65
9	60.60	9.23	0.0007	9.48

Table 4: This table reports the codebooks of neurons, credibility estimators of claims frequencies and relative sizes of clusters.

Table 4 provides detailed information about codebooks and estimated claims frequencies. Two neurons associated to younger insured cover 30% of the portfolio. Neurons 3 and 4 gather less 1.3% of policies and the frequency of claims associated to these risks is particularly high (71% and 12%) due to the lack of contracts in these clusters. A way to smooth frequencies of claims consists to increase the credibility parameters γ and/or to decrease the weight β. A comparison of Figures 5 and 3 allows us to visualize the impact of the Bayesian metric on the segmentation. For example, the owners of a motorcycle older than 30 years are assigned to the 6^{th} neuron in Table 4. Whereas a segmentation only based on owner's and vehicle ages divides this group into two clusters (neurons 3 and 8 in Table 2).

Algorithm 4 Bayesian regression Kohonen's algorithm for quantitative variables.

Initialization :

Randomly attribute codebooks $\boldsymbol{\omega}_1(0)$,..., $\boldsymbol{\omega}_{l^2}(0)$.

Set $\bar{\lambda}_u(0) = \bar{\lambda}$.

Main procedure :

For $e = 0$ to maximum epoch, e_{max}

For $i = 1$ to n

1) Find the BMN (best matching node) u matching the i^{th} policy

$$u = \arg\min_u d(\boldsymbol{x}_i, \boldsymbol{\omega}_u, \bar{\lambda}_u) .$$

2) Update $\Omega_u(e)$ and codebooks in the BMN neighborhood:

$$\boldsymbol{\omega}_v(e+1) = \boldsymbol{\omega}_v(e) + \theta(u, v, e)\, \epsilon(e)\, (\boldsymbol{x}_i - \boldsymbol{\omega}_v(e))$$

for $v = 1, ..., l^2$, with

$$\begin{aligned} \epsilon(e) &= \epsilon_0 \left(\frac{e_{max} - e}{e_{max}} \right) , \\ \theta(u, v, e) &= \theta_0 \exp\left(- \frac{(\|C_u - C_v\|_2)^2}{2\sigma(e)^2} \right) , \\ \sigma(e) &= \sigma_0 \left(\frac{1.2\, e_{max} - e}{e_{max}} \right) . \end{aligned}$$

End loop on policies, i

3) Update of $\bar{\lambda}_u(e+1) = \frac{\sum_{i \in \Omega^u(e)} N_i}{\sum_{i \in \Omega^u(e)} \nu_i}$ and $\alpha_u(e+1) = \frac{\nu^{\Omega_u(e)} \bar{\lambda}}{\gamma + \nu^{\Omega_u(e)} \bar{\lambda}}$.

4) Calculation of the total distance d^{total} between policies and BMNs:

$$d^{total} = \sum_{i=1}^{n} \left\| \boldsymbol{\omega}_{BMN(i)}(e+1) - \boldsymbol{x}_i \right\|_2 + \beta\, d_{\Omega_{BMN(i)}}(i, \bar{\lambda}_{BMN(i)}(e+1))$$

End loop on epochs, e

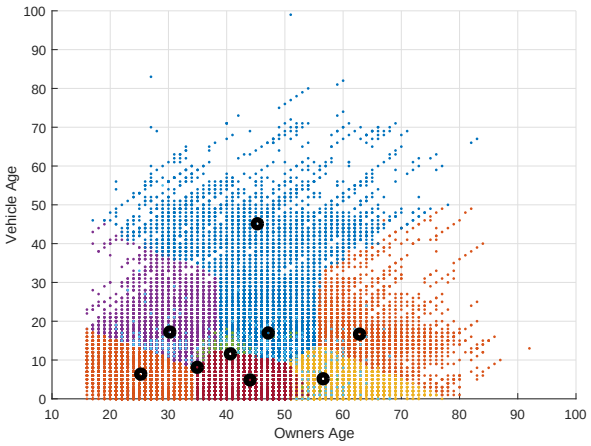

Figure 5: Regression Kohonen's map of the portfolio, with 9 neurons. The segmentation variables are the owner's age and age of the vehicle. Each policy is represented by a dot. Black dots point out the position of neurons. The colors identify the area of influence of neurons.

The total distance d^{total} measures the quality of the regression and of the portfolio segmentation, based on the distance in equation (13). If we aim at evaluating the goodness of fit, we may be tempted to calculate the deviance. However, using a Bayesian estimator perturbs the interpretation of this measure. The deviance is the difference between the log-likelihood of a model with as many parameters than observations and the fitted model. The model counting as many parameters than observations is called the *saturated model*. The log-likelihood of this model, denoted by $l^{saturated}$, is computed with parameters set to empirical frequency. The saturated model has no predictive power but provides the best fit from a pure statistical point of view given that it has the highest attainable log-likelihood. If we denote by l the log-likelihood for our data, the deviance D^* is defined as likelihood ratio test (LRT) of the model under consideration against the saturated model:

$$D^* = 2\left(l^{saturated} - l\right) .$$

If $\hat{\lambda}_i$ is the estimate of the claims frequency for the i^{th} policy forecast by the SOM, the probability of observing m claims over a period ν_i is equal to

$$P\left(N_i = m\right) = \frac{\left(\hat{\lambda}_i \nu^i\right)^m}{m!} e^{-\hat{\lambda}_i \nu^i} .$$

The contribution of the i^{th} policy to the log-likelihood l is then equal to

$$N_i \log\left(\hat{\lambda}_i \nu^i\right) - \hat{\lambda}_i \nu^i - N_i! .$$

In the saturated model, this contribution is given by

$$\begin{cases} N_i \log\left(\frac{N_i}{\nu^i}\nu^i\right) - \frac{N_i}{\nu^i}\nu^i - N_i! & if\ N_i \geq 1 \\ 0 & else\ if \end{cases}.$$

The marginal deviance for the i^{th} policy is then equal to

$$\begin{cases} 2N_i\left(\frac{\nu_i}{N_i}\hat{\lambda}_i - \log\frac{\hat{\lambda}_i\nu_i}{N_i} - 1\right) & if\ N_i \geq 1 \\ 2\nu_i\hat{\lambda}_i & if\ N_i = 0 \end{cases}.$$

The total deviance is the sum of marginal contributions

$$D^* = \ 2\sum_{i=1}^{n} N_i\left(\frac{\nu_i}{N_i}\hat{\lambda}_i - \left(\log\frac{\hat{\lambda}_i\nu_i}{N_i} + 1\right)I_{\{N_i\geq 1\}}\right).$$

The deviance is used in statistics to compare the predictive capacity of models fitted by log-likelihood maximization. This measure is however not fully adapted to compare regressive self-organizing maps in a Bayesian set-up. This point is emphasized by Table 5 that reports deviances and average distances between policies and best matching neurons, for different values of β and credibility weights γ. As revealed by Table 5, increasing the number of neurons (and then of clusters) reduces the average distance but raises the deviance. This counter-intuitive result is the direct consequence of using credibility weights that ensure the positivity of estimated frequencies. To understand this, let us consider an extreme case in which we have as many neurons than data and when β is small. In this case, each neuron is associated to a single policy and the frequency estimate is

$$\mathbb{E}\left(\lambda_i\Theta|N_i\right) = \alpha_i\bar{\lambda}_i + \left(1-\alpha_i\right)\bar{\lambda},$$

where $\bar{\lambda}_i = \frac{N_i}{\nu_i}$ and $\bar{\lambda} = \frac{\sum_{i=1}^{N} N_i}{\sum_{i=1}^{N}\nu_i}$. Given that the average intensity is low, $\bar{\lambda} = 0.0106$, the weight $\alpha_i = \left(1+\frac{\gamma}{\nu_i\bar{\lambda}}\right)^{-1}$ tends to zero and $\mathbb{E}\left(\lambda_i\Theta|N_i\right) \approx \bar{\lambda}$. This model has the same log-likelihood as the one obtained with a Poisson model with a single parameter. This model, called the *null model*, has the lowest log-likelihood and the highest deviance. Then, refining the segmentation does not necessary improve the deviance if we use a credibility estimator for frequencies. This point was underlined by Wüthrich and Buser (2017) who observe the same phenomenon for regression trees. On page 83, they mention that replacing the maximum likelihood estimator (MLE) by the empirical credibility estimator has the advantage that the resulting frequency estimators are always strictly positive. The disadvantage is that errors measured by the deviance may increase by adding more splits (which turns out to refine the segmentation). This is because only the MLE minimizes the corresponding deviance statistics and maximizes the log-likelihood function, respectively.

Neuron l^2	γ	β	D^*	d^{total}
4	4	10	659	10344
9	4	10	679	7829
16	4	10	694	7078
4	10	10	1097	11051
9	10	10	1252	8988
16	10	10	1167	8567
4	10	1	2160	6368
9	10	1	2347	4169
16	10	1	2519	3363
9	0	0	6087	3331
GLM			6214	
Null model			6648	

Table 5: This table reports deviances and average distances between policies and best matching neurons, for different values of β and credibility weights γ. We also provide the deviance of a non-regressive SOM, of a GLM model and the null model.

Estimated claims frequencies obtained with a credibility weight $\gamma = 4$ and reported in table 4, vary a lot. Increasing the credibility parameters γ and/or to decreasing the weight β allows to smooth these frequencies. However, figures of table 5 show that this smoothing deteriorates the deviance. To get an idea to which extend large deviances are admissible, we fit a generalized linear model (GLM) to our dataset. GLM is a standard approach widely used by insurers to assess their risk. We regress the claims frequency with respect to six categories of owner's age: (28;31], (31;35], (35;44], (44;51], (51;61] and (61;100]. We fit a Poisson model with a logarithmic link function. The residual deviance for this model climbs to 6214, which is quite high compared to deviances obtained with a regressive SOM. This confirms that regressive SOMs have comparable performances to GLM. Except that it is a non-parametric and non supervised approach.

In theory, it is possible to perform cross validation. In practice, we face several difficulties when we implement this method. Firstly, the number of claims being small (693 claims for 62436 contracts), some bootstrapped data samples do not contain any claims. Secondly, the cross validation is computationally time consuming.

2.4 Comparison of Bayesian SOM and k-means clustering method

this section compares the Bayesian regressive SOM to a Bayesian version of the k-means algorithm. We use the same notations as in subsection 2.2. We partition the dataset X of n records in $\mathbb{R}^p$ into k clusters. The coordinates of the u^{th} centroid are contained in a vector $\boldsymbol{c}_u = (c_1^u, \ldots, c_p^u)$ and S_u is the cluster of policies associated to $\boldsymbol{c}_u$ for $u = 1, ..., k$. As for the Bayesian SOM, we replace the Euclidian distance $d(\boldsymbol{x}_i, \boldsymbol{c}_u)$ in the Lloyd's algorithm

by

$$d(\boldsymbol{x}_i, \boldsymbol{c}_u, \bar{\lambda}_u) := \|\boldsymbol{c}_u - \boldsymbol{x}_i\|_2 + \beta\sqrt{\left(N_i - \mathbb{E}\left(\lambda_u\Theta|N^{S_u}\right)\nu_i\right)^2},$$

where $\beta \in \mathbb{R}^+$. N^{S_u} and ν^{S_u} are respectively the number of claims and the total exposure for the u^{th} cluster. $\mathbb{E}\left(\lambda_u\Theta|N^{S_u}\right)$ is the estimate of the claims frequency for S^u where Θ is distributed as a $\Gamma(\gamma,\gamma)$ random variable. As in Subsection 2.3 $\bar{\lambda}_u = \frac{N^{S_u}}{\sum_{\boldsymbol{x}_i \in S_u}\nu_i}$ and the observed empirical estimator of $\mathbb{E}\left(\lambda_u\Theta|N^{S_u}\right)$ is

$$\mathbb{E}\left(\lambda_u\Theta|N^{S_u}\right) \quad = \quad \alpha_u\bar{\lambda}_u + (1-\alpha_u)\bar{\lambda},$$

where $\alpha_u = \frac{\nu^{S_u}\bar{\lambda}}{\gamma+\nu^{S_u}\bar{\lambda}}$. Algorithm 5 details a variant of the k-means algorithm including the same Bayesian regressive feature than the SOM.

Algorithm 5 adapted Lloyd's algorithm for Bayesian regression.

Initialization:

Initialize positions of centroids $\boldsymbol{c}_1(0)$,...,$\boldsymbol{c}_k(0)$ with Algorithm 3.

Set $\bar{\lambda}_u(0) = \bar{\lambda}$ for $u = 1, ..., k$.

Main procedure:

For $e = 0$ to maximum epoch, e_{max}

Assignment step:

For $i = 1$ to n

1) Assign $\boldsymbol{x}_i$ to a cluster $S_u(e)$ where $u \in \{1, ..., k\}$

$$S_u(e) = \{\boldsymbol{x}_i \ : \ d(\boldsymbol{x}_i, \boldsymbol{c}_u(e), \bar{\lambda}_u(e)) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_j(e), \bar{\lambda}_j(e)) \ \forall j \in \{1, ..., k\}\}$$

End loop on policies, i.

Update step:

For $u = 1$ to k

2) Calculate the new centroids $\boldsymbol{c}_u(e+1)$ of $S_u(e)$ as follows

$$\boldsymbol{c}_u(e+1) = \frac{1}{|S_u(e)|}\sum_{\boldsymbol{x}_i \in S_u(e)} \boldsymbol{x}_i$$

3) Update of $\bar{\lambda}_u(e+1) = \frac{\sum_{\boldsymbol{x}_i \in S_u(e)} N_i}{\sum_{\boldsymbol{x}_i \in S_u(e)} \nu_i}$ and $\alpha_u = \frac{\nu^{S_u(e)}\bar{\lambda}}{\gamma+\nu^{S_u(e)}\bar{\lambda}}$.

End loop on centroids, u

4) Calculation of the total distance d^{total} between policies and centroids:

$$d^{total} = \sum_{u=1}^{k}\sum_{\boldsymbol{x}_i \in S_u(e)} d(\boldsymbol{x}_i, \boldsymbol{c}_u(e+1), \bar{\lambda}_u(e+1))$$

End loop on epochs e.

Table 6 reports the deviances and distances obtained after 100 iterations of the adapted Lloyd's algorithm for Bayesian regression. We use the same configurations as for the SOMs. In terms of distance, the SOM and K-mean algorithms achieve similar performance. Table 7 presents the coordinates of centroids and the Bayesian claims frequency per cluster. These results are computed with $\beta = 10$, $\gamma = 4$ and 9 clusters. A comparison of centroid positions with neural codebooks in Table 4 confirms that both algorithms gather insurance policies in a very similar way.

K-means k	γ	β	D^*	d^{total}
4	4	10	660	10333
9	4	10	659	7979
16	4	10	708	7038
4	10	10	1097	11050
9	10	10	1121	9010
16	10	10	1179	8500
4	10	1	2131	6430
9	10	1	2366	4190
16	10	1	2500	3364
9	0	0	6095	3338

Table 6: This table reports deviances and average distances between policies and centroids, for different values of β and credibility weights γ.

Centroid u	$c_u^1 \times$ $max(Age)$	$c_u^2 \times$ $max(Veh.\,Age)$	$\mathbb{E}\left(\lambda_u \Theta \vert N^{S_u}\right)$	% of the population
1	25.94	5.70	0.0007	17.35
2	29.50	16.68	0.0005	15.63
3	34.48	8.01	0.7166	0.91
4	40.14	7.18	0.1213	0.30
5	45.94	44.38	0.0016	4.13
6	46.07	16.90	0.0002	24.33
7	47.45	4.60	0.0002	21.08
8	58.76	2.06	0.0164	1.03
9	60.98	12.62	0.0004	15.24

Table 7: This table reports the positions of centroids, the Bayesian estimator of frequency of claims for clusters of policies associated to centroids, and sub-population relative sizes.

3 Analysis of qualitative variables with a SOM

Many features of insurance policies are described by categorical variables that cannot be handled with a classic SOM. In this section, we modify the Kohonen algorithm in order to analyze a dataset that exclusively contains this type of variables. We first propose a procedure to study the dependencies between variables. In section 5, this method is combined with the Bayesian SOM to regress claims frequency on quantitative and categorical variables.

First, we introduce the structure of data to which the algorithm is applied. The number of insurance policies is still denoted by n. Each of these policies is described by K variables which have m_k binary modalities for $k = 1, ..., K$. By binary, we mean that the modality j of the k^{th} variable is identified by an indicator variable equal to zero or one. The total number of modalities is the sum of m_k: $m = \sum_{k=1}^{K} m_k$. In further developments, we enumerate modalities from 1 to m. The information about the portfolio may be summarized by a $n \times m$ matrix $D = (d_{i,j})_{i=1...n, j=1...m}$. If the i^{th} policy presents the j^{th} modality then $d_{i,j} = 1$ and $d_{i,j} = 0$ otherwise.

For example, let us assume that a policy is described by the gender (M=male or F=Female) of the policyholder and by a geographic area (U=urban, S=suburban or C=countryside). The number of variables and modalities are respectively $K = 2$, $m_1 = 2$ and $m_2 = 3$. If the first and second policyholders are respectively a man living in a city and a woman living in the countryside, the two first lines of the matrix D are presented in table 8.

	Gender		Area		
Policy	M	F	U	S	C
1	1	0	1	0	0
2	0	1	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮

Table 8: Example of a disjunctive table for $K = 2$ variables with respectively $m_1 = 2$, $m_2 = 3$ modalities.

The table D is called a disjunctive table. In order to study the dependence between the modalities, we need to calculate the numbers $n_{i,j}$ of individuals sharing modalities i and j, for $i, j = 1, ..., m$. The $m \times m$ matrix $B = (n_{i,j})_{i,j=1,...,M}$ is a contingency table, called the Burt matrix containing this information. The Burt matrix is directly related to the disjunctive table as follows:

$$B = D^\top D\,.$$

This symmetric matrix is composed of $K \times K$ blocks $B_{k,l}$ for $k, l = 1, ..., K$. A block $B_{k,l}$ is the contingency table that crosses the variables k and l. Table 9 shows the Burt matrix for the matrix D presented in Table 8. By construction, the sum of elements of a block

$B_{k,l}$ is equal to the total number of policies, n. The sum of $n_{i,j}$ of the same row i is equal to

$$n_{i,.} = \sum_{j=1,...,m} n_{i,j} = K\, n_{i,i}\,.$$

The Burt matrix being symmetric, we directly infer that

$$n_{.,j} = \sum_{i=1,...,m} n_{i,j} = K\, n_{j,j}\,.$$

Furthermore, blocks $B_{k,k}$ for $k = 1, ..., K$ are diagonal matrix, whose diagonal entries are the numbers of policies who respectively present the modalities $1, ..., m_k$, for the variable k . In our example, we have that $n_{1,1} + n_{2,2} = n$ and $n_{3,3} + n_{4,4} + n_{5,5} = n$. $n_{1,1}$ and $n_{2,2}$ count the total number of men and women in the portfolio. Whereas $n_{3,3}$, $n_{4,4}$ and $n_{5,5}$ counts the number of policyholders living respectively in a urban, sub-urban or rural environment.

		Gender		Area		
		M	F	U	S	C
Gender	M	$n_{1,1}$	0	$n_{1,3}$	$n_{1,4}$	$n_{1,5}$
	F	0	$n_{2,2}$	$n_{2,3}$	$n_{2,4}$	$n_{2,5}$
Area	U	$n_{3,1}$	$n_{3,2}$	$n_{3,3}$	0	0
	S	$n_{4,1}$	$n_{4,2}$	0	$n_{4,4}$	0
	C	$n_{5,1}$	$n_{5,2}$	0	0	$n_{5,5}$

Table 9: Burt matrix for the disjunctive Table 8.

The self-organizing map requires the definition of a distance between categorical variables. This point is discussed in the next section.

4 A χ^2 distance for categorical variables

Multiple correspondence Analysis (MCA) was initially developed by Burt (1950) and enhanced by Benzécri (1973), Greenacre (1984) and Lebart et al.(1984). This technique evaluates the level of dependence between categorical variables with a χ^2 distance. We first present this distance in the case of two variables and extend it later to the multivariate case.

4.1 The bivariate case

Let us first consider the case of two categorical variables, $K = 2$, with m_1 and m_2 modalities. The classical MCA studies relative frequencies of crossed modalities. The table of frequency $F = (f_{i,j})_{i,j}$ is a $m_1 \times m_2$ matrix defined as follows:

$$f_{i,j} \quad = \quad \frac{n_{i,j}}{n} \quad i = 1, ..., m_1\,,\; j = 1, ..., m_2\,.$$

The marginal frequencies are equal to

$$\begin{aligned} f_{i,.} &= \sum_{j=1}^{m_2} f_{i,j} = \frac{n_{i,.}}{n} \quad i = 1, ..., m_1 \,, \\ f_{.,j} &= \sum_{i}^{m_1} f_{i,j} = \frac{n_{.,j}}{n} \quad j = 1, ..., m_2 \,. \end{aligned}$$

If variables are independent, the expected number of policies with modalities i and j, is equal to $\tilde{n}_{i,j} = \frac{n_{i,.}\, n_{.,j}}{n}$. In this case, standardized residuals $\chi_{i,j} = \frac{n_{i,j} - \tilde{n}_{i,j}}{\sqrt{\tilde{n}_{i,j}}}$ should be $N(0,1)$ random variables. If the two variables are independent, then the following statistics (called the inertia):

$$\begin{aligned} \chi^2 &= \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \frac{(n_{i,j} - \tilde{n}_{i,j})^2}{\tilde{n}_{i,j}} \\ &= n \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \frac{(f_{i,j} - f_{i,.} f_{.,j})^2}{f_{i,.} f_{.,j}} \\ &= n \left(\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \frac{f_{i,j}^2}{f_{i,.} f_{.,j}} - 1 \right) , \end{aligned}$$

is a chi-square random variable with $(m_1 - 1)(m_2 - 1)$ degrees of freedom. This justifies to measure the distance between the modalities i and i' of the first variable by

$$\begin{aligned} \chi^2(i, i') &= \sum_{j=1}^{m_2} \frac{1}{f_{.,j}} \left(\frac{f_{i,j}}{f_{i,.}} - \frac{f_{i',j}}{f_{i',.}} \right)^2 , \\ &= \sum_{j=1}^{m_2} \frac{n}{n_{.,j}} \left(\frac{n_{i,j}}{n_{i,.}} - \frac{n_{i',j}}{n_{i',.}} \right)^2 . \end{aligned} \tag{14}$$

$\chi^2(i, i')$ is the distance between the rows i and i' of the matrix of frequency F. It may be checked that the dispersion of rows around their barycenter is equal to the inertia. Similarly, the chi-square distance between the modalities j and j' of the second variable is defined by

$$\begin{aligned} \chi^2(j, j') &= \sum_{i=1}^{m_1} \frac{1}{f_{i,.}} \left(\frac{f_{i,j}}{f_{.,j}} - \frac{f_{i,j'}}{f_{.,j'}} \right)^2 \\ &= \sum_{i=1}^{m_1} \frac{n}{n_{i,.}} \left(\frac{n_{i,j}}{n_{.,j}} - \frac{n_{i,j'}}{n_{.,j'}} \right)^2 . \end{aligned} \tag{15}$$

This measures the distance between columns of the matrix F. In the SOM algorithm, we prefer to evaluate distances between rows and columns with the Euclidian distance. It is then convenient to replace the frequencies $f_{i,j}$ by weighted values $f_{i,j}^W$:

$$f_{i,j}^W := \frac{f_{i,j}}{\sqrt{f_{i,.}\, f_{.,j}}} = \frac{n_{i,j}}{\sqrt{n_{i,.}\, n_{.,j}}} \quad i = 1, ..., m_1 \; j = 1, ..., m_2 \,. \tag{16}$$

The distances between rows (i, i') and columns (j, j') simplify as follows:

$$\chi^2\left(i, i'\right) = \sum_{j=1}^{m_2} \left(f_{i,j}^W - f_{i',j}^W\right)^2 .$$
$$\chi^2\left(j, j'\right) = \sum_{i=1}^{m_1} \left(f_{i,j}^W - f_{i,j'}^W\right)^2 .$$

Finally, notice that the matrix $F^W = \left(f_{i,j}^W\right)_{i=1,...,m_1, j=1,...,m_2}$ is symmetric by construction.

4.2 The multivariate case

In this section, we extend the notion of χ^2 distance between categorical variables to the multivariate case ($K > 2$). Remember that the Burt table, $B = (n_{i,j})_{i,j=1,...,m}$, is a contingency table. This symmetric matrix is composed of $K \times K$ blocks $B_{k,l}$. $B_{k,l}$ is itself the contingency table crossing variables k and l. It is then natural to extend the definition of distance (14) between rows i and i' of the Burt matrix as follows:

$$\chi^2\left(i, i'\right) = \sum_{j=1}^{m} \frac{n}{n_{.,j}} \left(\frac{n_{i,j}}{n_{i,.}} - \frac{n_{i',j}}{n_{i',.}}\right)^2 \quad i, i' \in \{1, ..., m\}.$$

Similarly, the chi-square distance between columns j and j' of the Burt matrix is defined by

$$\chi^2\left(j, j'\right) = \sum_{i=1}^{m} \frac{n}{n_{i,.}} \left(\frac{n_{i,j}}{n_{.,j}} - \frac{n_{i,j'}}{n_{.,j'}}\right)^2 \quad j, j' \in \{1, ..., m\}.$$

As we prefer to evaluate distances with the Euclidian distance, the elements of the Burt matrix $n_{i,j}$ are replaced by weighted values $n_{i,j}^W$:

$$n_{i,j}^W := \frac{n_{i,j}}{\sqrt{n_{i,.}\, n_{.,j}}} \quad i, j = 1, ..., m \,. \tag{17}$$

Given that the sums $n_{i,.} = K\, n_{i,i}$ and $n_{.,j} = K\, n_{j,j}$, we have that .

$$n_{i,j}^W := \frac{n_{i,j}}{K\sqrt{n_{i,i}\, n_{j,j}}} \quad i, j = 1, ..., m \,. \tag{18}$$

If C is the diagonal matrix $C = diag\left(n_{11}^{-\frac{1}{2}}..n_{mm}^{-\frac{1}{2}}\right)$ then the weighted Burt matrix is denoted by B^W:

$$B^W = \frac{1}{K} C\, B\, C\,.$$

The distances between rows (i, i') and columns (j, j') of the Burt matrix becomes:

$$\chi^2\left(i, i'\right) = \sum_{j=1}^{m_2} \left(n_{i,j}^W - n_{i',j}^W\right)^2\,,$$
$$\chi^2\left(j, j'\right) = \sum_{i=1}^{m_1} \left(n_{i,j}^W - n_{i,j'}^W\right)^2\,.$$

4.3 Application to insurance data

Each modality of categorical variables is represented by a line of the weighted Burt matrix. A line of this matrix defines a point in a space with m dimensions. The level of dependence between modalities i and j is measured by the Euclidian distance between two points with coordinates contained in the i^{th} and j^{th}lines of B^W. Therefore, we apply the Kohonen's algorithm 1 directly to the weighted Burt matrix in order to study the relations between categorical variables. We consider the categorical variables described in table 1 to which we add a new categorical variable representative of the owner's age. We consider three modalities for the owner's age, that are constructed according to the rule reported in Table 10.

Discretized Age categories	
Modality	Value
Young	Owners Age$<$ 35 years old
Mature	35 years old $\leq$ Owners Age$<$ 55 years old
Old	55 years old $\leq$ Owners Age

Table 10: Table of modalities for the categorical variable representative of the owner's age.

We fit a 4×4 SOM to the matrix B^W. The number of epochs is set to $e_{max} = 1000$. The initial codebooks are equal to randomly drawn lines of B^W. The parameters of equations (3), (4) and (5) for the update of codebooks are: $\epsilon_0 = 0.01$, $\theta_0 = 1$ and $\sigma_0 = 0.10$. The network is trained after 850 iterations and 6 neurons are not assigned to any modality. One neuron is coupled to 4 modalities and two neurons each regroup 3 modalities. Details about the pooling of modalities are provided in Table 11. This reveals that the recurrent insured profile is a mature man living in the countryside and owning a vehicle of class 3. We also learn that a majority of young insureds lives in small cities whereas insured women are living in a urban environment and drive a motorcycle of class 4. The older policyholders drives a vehicle of class 2 whereas people living in northern cities or Gotland

mainly own a powerful motorcycle (class 7). This analysis reveals that SOMs detect the most common associations of modalities. With this information, we can draw a composite drawing of the average policy.

Neuron u	Modalities	Marginal frequency	Neuron u	Modalities	Marginal frequency
1	Mature	0.006	9	Class 5	0.011
	Men	0.011	10	Suburban	0.016
	Countryside	0.006	12	North. village	0.006
	Class 3	0.008	13	Old	0.005
2	Young	0.027	13	Class 2	0.014
	Small city	0.010	15	Class 6	0.020
4	Women	0.009	16	North. city	0.006
	Urban	0.029	16	Gotland	0.004
	Class 4	0.008	16	Class 7	0.018
7	Class 1	0.009			

Table 11: Groups of modalities per neuron. We also report the average claims frequencies for each of these modalities.

Number of Centroids	K-means d^{total}	Number of Neurons	SOM d^{total}
2	17.25	4	15.80
4	16.63	9	11.30
6	16.01	16	10.16
8	14.89		
10	15.57		
12	15.47		
14	16.04		
16	16.73		

Table 12: Total distances d^{total} between policies-centroids and policies-codebooks of best matching neurons.

In Subsections 2.2 and 2.3, we emphasize that SOM and K-means algorithm achieve similar performances at least when the number of data per contract is small (in our case, the owner and vehicle ages) and when the dataset is large (62 436 policies). Here, the pooling of categorical variables is done by analyzing a small dataset with a comparatively high number of modalities per data. In our case study, the Burt Matrix B^W counts only 19 observations and the same number of modalities by construction. Table 12 reveals that in this particular context, the K-means algorithm converges to a local minimum in term of total distance. The SOM finds a better solution whatever the configuration. Contrary

to the SOM, the K-means algorithm even becomes unstable when the number of centroids increase.

Algorithm 6 Bayesian Regression Kohonen's algorithm for quantitative and categorical variables.

Initialization :

Randomly attribute codebooks $\boldsymbol{q}_u(0)$ and $\boldsymbol{\omega}_u(0)$, $u = 1, ..., l^2$.

Set $\bar{\lambda}_u = \bar{\lambda} = \frac{\sum_{u=1}^{l^2} N^{\Omega_u}}{\sum_{u=1}^{l^2} \nu^{\Omega_u}}$.

Main procedure :

For $e = 0$ to maximum epoch, e_{max}

For $i = 1$ to n

1) Find the BMN u matching the i^{th} policy

$$u = \arg\min_u d(\boldsymbol{x}_i, D_{i,.}, \boldsymbol{q}_u, \boldsymbol{\omega}_u, \bar{\lambda}_u) .$$

2) Update codebooks the neighborhood of the BMN

$$\boldsymbol{q}_v(e+1) = \boldsymbol{q}_v(e) + \theta(u, v, e)\, \epsilon(e)\, (\boldsymbol{x}_i - \boldsymbol{q}_v(e))$$
$$\boldsymbol{\omega}_v(e+1) = \boldsymbol{\omega}_v(e) + \theta(u, v, e)\, \epsilon(e)\, \left(D_{i,.} B^W / K - \boldsymbol{\omega}_v(e)\right)$$

where

$$\epsilon(e) = \epsilon_0 \left(\frac{e_{max} - e}{e_{max}}\right) ,$$
$$\theta(u, v, e) = \theta_0 \exp\left(-\frac{(\|C_u - C_v\|_2)^2}{2\sigma(e)^2}\right) ,$$
$$\sigma(e) = \sigma_0 \left(\frac{1.2\, e_{max} - e}{e_{max}}\right) .$$

End loop on policies, i.

3) Calculation of the distance d^{total} between policies and BMNs:

$$d^{total} = \sum_{i=1}^{n} \left\| \boldsymbol{q}_{BMN(i)} - \boldsymbol{x}_i \right\|_2 + \beta_1 \left\| \boldsymbol{\omega}_{BMN(i)} - D_{i,.} B^W / K \right\|_2 + \beta_2\, d_{\Omega_{BMN(i)}}(i, \bar{\lambda}_{BMN(i)})$$

4) Update of $\bar{\lambda}_u = \frac{\sum_{i \in \Omega^u} N_i}{\sum_{i \in \Omega^u} \nu_i}$ for $u = 1, ..., l^2$.

End loop on epochs, e.

---

5 A regressive SOM with quantitative and categorical variables

In this section, we exploit information contained in categorical and quantitative variables for regressing claims frequencies. The n policies are described by p quantitative and K categorical variables. The categorical variables have m_k binary modalities for $k = 1, ..., K$. The total number of modalities is denoted $m = \sum_{k=1}^{K} m_k$. The information about the portfolio is summarized by the $m \times m$ weighted Burt matrix B^W. Whereas the feature of each policies are reported in the $n \times m$ disjunctive table D. The quantitative variables stored in a table X with n rows and l columns. The neural map is a $l \times l$ array of neurons with a matrix C that contains the coordinates of nodes in $[0,1] \times [0,1]$. Neurons are equally spaced on this pavement. The codebook of the u^{th} neurons is now composed of a $p-$vector

$$\boldsymbol{q}_u = \left(q_1^u, ..., q_p^u\right)$$

for quantitative variables and of a m-vector

$$\boldsymbol{\omega}_u \quad = \quad (\omega_1^u, ..., \omega_m^u)$$

for qualitative variables. The distance between the quantitative variables of the i^{th} policy and the neuron is measured by the 2 norm : $\|\boldsymbol{q}_u - \boldsymbol{x}_i\|_2$. The modalities of categorical variables are represented by points in a space with m dimensions and their coordinates are contained in weighted Burt matrix. It is then natural to measure the distance between the modalities of the i^{th} policy and the neuron by the following norm:

$$\left\|\boldsymbol{\omega}_u - D_{i,\cdot} B^W / K\right\|_2 , \tag{19}$$

where $D_{i,\cdot}$ is the i^{th} line of the disjunctive table. The quantity (19) is the distance between the neuron represented by a point of coordinates W_u in $\mathbb{R}^m$ and the barycenter $D_{i,\cdot} B^W / K$ of the cloud of points corresponding to modalities characterizing the i^{th} policy. As in previous section we denote by $\Omega_{u=1,...,l^2}$ the set of insurance contracts assigned to the u^{th} neuron. The credibility estimator of the expected number of claims caused by a policy in Ω_u is equal to:

$$\mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) = \alpha_u \bar{\lambda}_u + (1 - \alpha_u)\, \bar{\lambda}\,,$$

where $\bar{\lambda}_u = \frac{N^{\Omega_u}}{\nu^{\Omega_u}}$ and $\bar{\lambda} = \frac{\sum_{u=1}^{l^2} N^{\Omega_u}}{\sum_{u=1}^{l^2} \nu^{\Omega_u}}$ is the global average claims frequency. The credibility weights are proportional to the total exposure in Ω_u:

$$\alpha_u = \frac{\nu^{\Omega_u} \bar{\lambda}}{\gamma + \nu^{\Omega_u} \bar{\lambda}} \qquad u = 1, ..., l^2\,.$$

The homogeneity of claims frequency inside a subset Ω_u is measured by the realized standard deviation of N^{Ω_u}:

$$d_{\Omega_u}(\bar{\lambda}_u) \quad = \quad \sqrt{\sum_{i \in \Omega_u} \left(N_i - \mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) \nu_i\right)^2}$$

and the distance between the realized frequency of the i^{th} policy and expected posterior distribution for the subgroup Ω_u is given by

$$d_{\Omega_u}(i, \bar{\lambda}_u) \quad = \quad \sqrt{\left(N_i - \mathbb{E}\left(\lambda_u \Theta | N^{\Omega_u}\right) \nu_i\right)^2} \, .$$

We use this standard deviation to define a new metric to measure the distance between the i^{th} policy and the u^{th} neuron:

$$\begin{aligned} d(\boldsymbol{x}_i, D_{i,\cdot}, \boldsymbol{q}_u, \boldsymbol{\omega}_u, \bar{\lambda}_u) := \|\boldsymbol{q}_u - \boldsymbol{x}_i\|_2 + \beta_1 \left\|\boldsymbol{\omega}_u - D_{i,\cdot} B^W / K\right\|_2 \\ + \beta_2 \, d_{\Omega_u}(i, \bar{\lambda}_u) \qquad u = 1, ..., l^2 \; i = 1, ..., n \end{aligned} \tag{20}$$

where β_1 and β_2 are weights adjusting the regressive feature of the map with respect to its segmentation function. Algorithm 6 presents the procedure to build the Bayesian regressive SOM with this distance. Notice that the initialization step can be done with Algorithm 3 if the convergence is too slow.

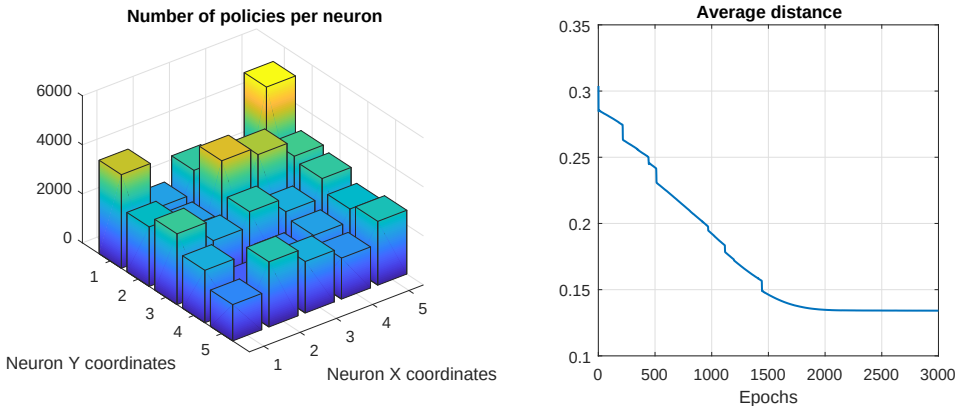

Figure 6: The left graph shows the number of policies assigned to each neuron. The right graph plots the evolution of the average distance $\frac{1}{n} d^{total}$ over iterations.

5.1 Application to insurance data

We fit a 5×5 regression SOM to our insurance data set. We consider the three categorical variables described in table 1 and the two quantitative variables: owner's age and vehicle age. Quantitative variables are scaled on the interval $[0, 1]$ as done in equations (6). We perform 3000 iterations for 9 up to 25 neurons. Parameters for the update of codebooks are: $\epsilon_0 = 0.01$, $\theta_0 = 1$ and $\sigma_0 = 0.10$. The weights β_1 and β_2 involved in the definition of the distance (20) are set to one. In order to smooth the claims frequencies for different type of policies, we choose a credibility parameter $\gamma = 10$. The left plot of Figure 6 presents the distribution of policies in the neural grid. The neuron regrouping the highest number of policies (8.2% of the dataset), regroups mainly mature men, living in the countryside

and driving a class 3 vehicle. In Section 2.3, this set of features was identified as dominant in the portfolio. The right graph of Figure 6 shows the evolution of the average distance between the features of a policy and the codebooks of the best matching neuron ($\frac{1}{n}d^{total}$). Convergence is achieved after 2000 iterations.

Number of neurons	Deviance	d^{total}
9	5865	29028
16	6161	25237
25	6256	18024

Table 13: Deviances and total distances computed with the SOM for 9, 16 and 25 neurons.

As mentioned in sub-section 2.3, the deviance is not necessary adapted to compare regressive self-organizing maps in a Bayesian set-up. This point is confirmed in Table 13: segmenting the portfolio does not improve the deviance due to the use of Bayesian estimator for claims frequencies. We compare the performance the SOM to GLM by fitting a Poisson model, with a logarithmic link function. We use as covariates: the gender, the area, the class of the vehicle, six categories of owner's age ((28;31], (31;35], (35;44], (44;51], (51;61] and (61;100]) and two categories of vehicle age ((0;10] , (10;100]). The deviances for the GLM and null models are respectively equal to 5775 and 6648. In term of deviances, the SOM provides a less good fit than a GLM. However this conclusion must be nuanced given that SOMs are totally unsupervised and non parametric methods. The deviance of SOM may also be adjusted by modifying the credibility parameter γ. Furthermore, as shown in sub-section 2.3, a model with a small deviance does not necessary provides a satisfactory segmentation of the portfolio.

The most interesting information for an insurer is reported in Table 14 that presents the dominant modalities associated to each neuron. The j^{th} modality of the k^{th} qualitative variable is dominant for the u^{th} neuron if it minimizes the distance between the codebook $\boldsymbol{\omega}_u$ and the lines of B^W corresponding to the m_k categories of the k^{th} variable:

$$j = \arg \min_{v \in \{1,..,m_k\}} \left\| \boldsymbol{\omega}_u - B^W_{\sum_{d=1}^{k-1} m_d + v,.} \right\|_2 . \tag{21}$$

For quantitative variables, the average owner's and vehicle ages of contracts assigned to the u^{th} neuron are respectively equal to $q_1^u \times max(Age)$ and $q_2^u \times max(Veh.Age)$. Among the dominant profiles, we retrieve categories (Men, Countryside, Class 3), (Men, Countryside, Class 4) and (Men, Urban, Class 4). Policies assigned to these three neurons represents respectively 8.2%, 6.39% and 6.12% of the population. Only five neurons are associated to women. The reason is that the population of female drivers is under-represented in the portfolio (9843 women on 62 436 contracts).

u	Gender	Area	Type	Age	Veh. Age	Freq.	Real Freq	% of pop.
1	Men	Ctryside	Class3	23.87	8.26	0.0421	0.066	3.37
2	Women	Suburban	Class3	27.32	11.31	0.0123	0.0135	3.43
3	Men	Smlcity	Class6	38.94	9.76	0.0161	0.0194	2.38
4	Men	Ctryside	Class6	40.51	10.33	0.0106	0.0106	4.25
5	Women	Ctryside	Class5	40.65	13.11	0.0063	0.0038	3.27
6	Men	Suburban	Class5	41.56	11.04	0.0167	0.0199	2.97
7	Men	Urban	Class4	41.62	10.4	0.0226	0.0267	6.12
8	Men	Smlcity	Class5	42.42	11.98	0.0125	0.0133	3.34
9	Men	Suburban	Class4	42.58	11.27	0.0124	0.0134	3.05
10	Men	Urban	Class3	42.62	10.23	0.0211	0.0254	3.89
11	Men	Ctryside	Class5	43.01	12.03	0.0059	0.0052	5.88
12	Women	Ctryside	Class3	43.22	9.78	0.0076	0.0065	4.62
13	Men	Northcity	Class3	43.7	10.95	0.0075	0.0053	2.93
14	Men	Ctryside	Class4	43.87	12.71	0.0057	0.0049	6.39
15	Men	Suburban	Class6	44.04	10.22	0.0216	0.027	3.79
16	Men	Suburban	Class3	44.73	10.81	0.0091	0.0086	4.58
17	Men	Smlcity	Class4	44.88	11.24	0.0093	0.0089	4.59
18	Men	Ctryside	Class2	44.89	14.67	0.0101	0.0099	3.19
19	Men	Smlcity	Class3	45.04	11.07	0.0068	0.0057	4.86
20	Men	Ctryside	Class1	45.41	23.9	0.007	0.0054	4.17
21	Men	Ctryside	Class3	45.47	12.49	0.0038	0.0031	8.2
22	Women	Ctryside	Class6	45.67	10.12	0.0159	0.0237	1.42
23	Men	Smlcity	Class1	46.55	38.12	0.0058	0.0023	2.68
24	Women	Smlcity	Class5	47.41	11.29	0.0081	0.0058	2.37
25	Men	Northvlge	Class3	47.69	12.53	0.0059	0.0038	4.26

Table 14: This table reports the dominant features for each neurons, estimated claims frequency and percentages of the insured population assigned to each neuron.

5.2 Comparison with the K-means algorithm

We use the same notations as in Subsection 2.4. We partition the dataset X of n records with p quantitative and K categorical variables, into k clusters. The information about categories is summarized by the $m \times m$ weighted Burt matrix B^W and features of policies are reported in the $n \times m$ disjunctive table D. The quantitative variables stored in a table X with n rows and l columns. As in the SOM algorithm, the coordinates of the u^{th} centroid of quantitative variables are contained in a p-vector $\boldsymbol{c}_u = (c_1^u, \ldots, c_p^u)$ and in a m-vector $\boldsymbol{v}_u = (v_1^u, \ldots, v_m^u)$ for qualitative variables. S_u is the cluster of policies associated to $(\boldsymbol{c}_u, \boldsymbol{v}_u)$ for $u = 1, ..., k$.

Algorithm 7 adapted Lloyd's algorithm for Bayesian regression with quantitative and categorical variables.

Initialization:

Initialize centroids $\boldsymbol{c}_1(0)$,...,$\boldsymbol{c}_k(0)$ and $\boldsymbol{v}_1(0)$,...,$\boldsymbol{v}_k(0)$ with Algorithm 3.

Set $\bar{\lambda}_u(0) = \bar{\lambda}$ for $u = 1, ..., k$.

Main procedure:

For $e = 0$ to maximum epoch, e_{max}

Assignment step:

For $i = 1$ to n

1) Assign $\boldsymbol{x}_i$ to a cluster $S_u(e)$ where $u \in \{1, ..., k\}$

$$S_u(e) = \{(\boldsymbol{x}_i, D_{i,;}) \, : \, d(\boldsymbol{x_i}, D_{i,;}, \boldsymbol{c}_u(e), \boldsymbol{v}_u(e), \bar{\lambda}_u(e)) \leq d(\boldsymbol{x}_i, D_{i,;}, \boldsymbol{c}_j(e), \boldsymbol{v}_j(e), \bar{\lambda}_j(e)) \; \forall j \in \{1, ..., k\}\}$$

End loop on policies, i.

Update step:

For $u = 1$ to k

2) Calculate the new centroids $\boldsymbol{c}_u(e+1)$ and $\boldsymbol{v}_u(e+1)$

$$\boldsymbol{c}_u(e+1) = \frac{1}{|S_u(e)|} \sum_{\boldsymbol{x}_{i.} \in S_u(e)} \boldsymbol{x}_i$$

$$\boldsymbol{v}_u(e+1) = \frac{1}{|S_u(e)|} \sum_{D_{i,.} \in S_u(e)} D_{i,.} B^W / K$$

3) Update $\bar{\lambda}_u(e+1) = \frac{\sum_{\boldsymbol{x}_i \in S_u(e)} N_i}{\sum_{\boldsymbol{x}_i \in S_u(e)} \nu_i}$ and $\alpha_u = \frac{\nu^{S_u(e)} \bar{\lambda}}{\gamma + \nu^{S_u(e)} \bar{\lambda}}$.

End loop on centroids, u

4) Calculation of the total distance d^{total} :

$$d^{total} = \sum_{u=1}^{k} \sum_{\boldsymbol{x}_i \in S_u(e)} d(\boldsymbol{x}_i, D_{i,.}, \boldsymbol{c}_u(e+1), \boldsymbol{v}_u(e+1), \bar{\lambda}_j(e+1))$$

End loop on epochs e.

We replace the distance $d(\boldsymbol{x}_i, \boldsymbol{c}_u, \bar{\lambda}_u)$ in the Bayesian version of the Lloyd's algorithm by

$$d(\boldsymbol{x}_i, D_{i,.}, \boldsymbol{c}_u, \boldsymbol{v}_u, \bar{\lambda}_u) := \|\boldsymbol{c}_u - \boldsymbol{x}_i\|_2 + \beta_1 \left\|\boldsymbol{v}_u - D_{i,.} B^W / K\right\|_2 + \beta_2 \sqrt{\left(N_i - \mathbb{E}\left(\lambda_u \Theta | N^{S_u}\right) \nu_i\right)^2},$$

where β_1, $\beta_2 \in \mathbb{R}^+$ are parameters tuning the regressive feature of the map. N^{S_u} and ν^{S_u} are respectively the number of claims and the total exposure for the u^{th} cluster.

$\mathbb{E}\left(\lambda_u \Theta | N^{S_u}\right)$ is the Bayesian estimate of the claims frequency for S^u as defined in subsection 2.4. Algorithm 7 details the variant of the k-means algorithm including the same features than the SOM.

u	Gender	Area	Type	Age	Veh. Age	Freq.	Real Freq	% of pop.
1	Men	Ctryside	Class4	24.91	7.88	0.0371	0.0572	3.1
2	Men	Suburban	Class6	27.75	9.41	0.0181	0.0699	2
3	Women	Ctryside	Class3	28.83	11.48	0.0091	0.0108	6.5
4	Men	Ctryside	Class3	31.75	13.22	0.0093	0.0068	3.4
5	Men	Urban	Class5	38.93	10.49	0.0371	0.038	1.83
6	Men	Smlcity	Class6	38.94	9.72	0.0067	0.0193	2.39
7	Men	Ctryside	Class6	40.53	10.31	0.0125	0.0106	4.25
8	Men	Smlcity	Class5	42.41	11.96	0.0074	0.0133	3.31
9	Men	Suburban	Class4	42.61	11.26	0.0147	0.0135	3.04
10	Men	Ctryside	Class5	43.02	12.04	0.0238	0.0052	5.88
11	Men	Smlcity	Class4	43.26	11.62	0.0097	0.0086	3.69
12	Men	Ctryside	Class5	43.44	12.61	0.0073	0.0088	1.88
13	Men	Ctryside	Class4	43.88	12.69	0.0057	0.0049	6.39
14	Men	Urban	Class3	43.99	10.43	0.0065	0.02	6.79
15	Men	Suburban	Class5	44.07	10.8	0.0059	0.0169	4.51
16	Men	Suburban	Class1	44.26	40.16	0.0078	0.0068	1.88
17	Men	Suburban	Class3	44.72	10.81	0.0028	0.0086	4.58
18	Men	Ctryside	Class2	44.73	14.73	0.0124	0.0104	3.19
19	Men	Smlcity	Class3	45.19	10.87	0.0161	0.0055	5.51
20	Men	Ctryside	Class1	45.37	23.88	0.0152	0.0058	4.17
21	Men	Smlcity	Class1	46.77	19.25	0.0105	0.0052	2.87
22	Men	Northvlge	Class3	47.35	11.95	0.0079	0.0039	2.7
23	Women	Ctryside	Class3	48.43	10.39	0.0088	0.0071	8.47
24	Men	Suburban	Class6	49.49	10.13	0.0106	0.0173	2.59
25	Men	Ctryside	Class3	55.1	12.03	0.0108	0.0017	5.08

Table 15: This table reports the dominant features for each centroids, estimated claims frequency and percentages of the insured population assigned to each centroids.

Table 15 that presents the dominant modalities associated to 25 centroids, calculated with equation (21) in which $\boldsymbol{\omega}_u$ is replaced by $\boldsymbol{v}_u$. Compared to dominant features identified by the SOM in Table 14, we retrieve similar clusters. E.g. both algorithm have neurons or centroids with dominant features corresponding to a male driver of class 4 vehicles and living in the countryside. However we observe small discrepancies between forecasts of claims frequencies for similar dominant profiles.

Number of centroids	Deviance	d^{total}
9	5848	29204
16	6082	24150
25	6216	18772

Table 16: Deviances and total distances computed by the k-means algorithm with 9, 16 and 25 centroids.

Table 16 reports the deviances and total distances obtained with the k-means algorithm for 9, 16 and 25 centroids. A comparison with Table 13 reveals that SOMs converge to solutions with a lower total distance than the k-means algorithm when we partition the dataset in 9 or 25 clusters. Whereas the k-means algorithm slightly outperforms the SOM in the configuration with 16 centroids. We also observe that the k-means algorithm converges to a solution in less than 100 iterations while the SOM needs around 2000 iterations. The SOM seems more robust than k-means but is more time consuming.

5.3 Regression with the SOM

The SOM does not only segment the portfolio, it also allows for regressing the claims frequency on explanatory variables. We create a matrix X filled with owner's and vehicle ages of policies for which we want to assess the claims frequency. Next, we construct the disjunctive table D containing the qualitative modalities of these policies. Finally, we identify the best matching neuron and report the claims frequency of this cluster. We plot in figure 7 forecast claims frequencies for different risk profiles and owner's ages. The vehicle age is constant and set to one year.

We draw several interesting conclusions from Figure 5. Men living in a urban environment have a higher probability of accident with a class 3 motorcycle than men living in the countryside driving the same vehicle. The claims frequency for men from northern villages is higher up to 28 years old. Contrary to men, female drivers living in a city and in the countryside share the same claims frequency for ages above 33 years. Female drivers of a class 3 motorcycle in the countryside have a higher probability of reporting a claim than men of the same area. More surprising, women older than 18 years, driving the most powerful vehicle (class 7) have a smaller claims frequency than class 3 drivers. However, these results must be nuanced given that women are under-represented in the portfolio (15.76% of the data set). Male drivers of a class 7 motorcycle in the countryside have a higher claims frequency than class 3 drivers but the frequency falls above 35 years old. On average, we also expect less claims for persons living in a small city than insureds in a suburban environment. Finally, female drivers of a powerful vehicle cause less accident than male drivers, in the countryside. For men, this frequency falls around 35 years old whereas for women, the claims frequency falls around 19 years old. These results confirm that SOM may be use for segmenting policyholders and forecasting the claims frequency.

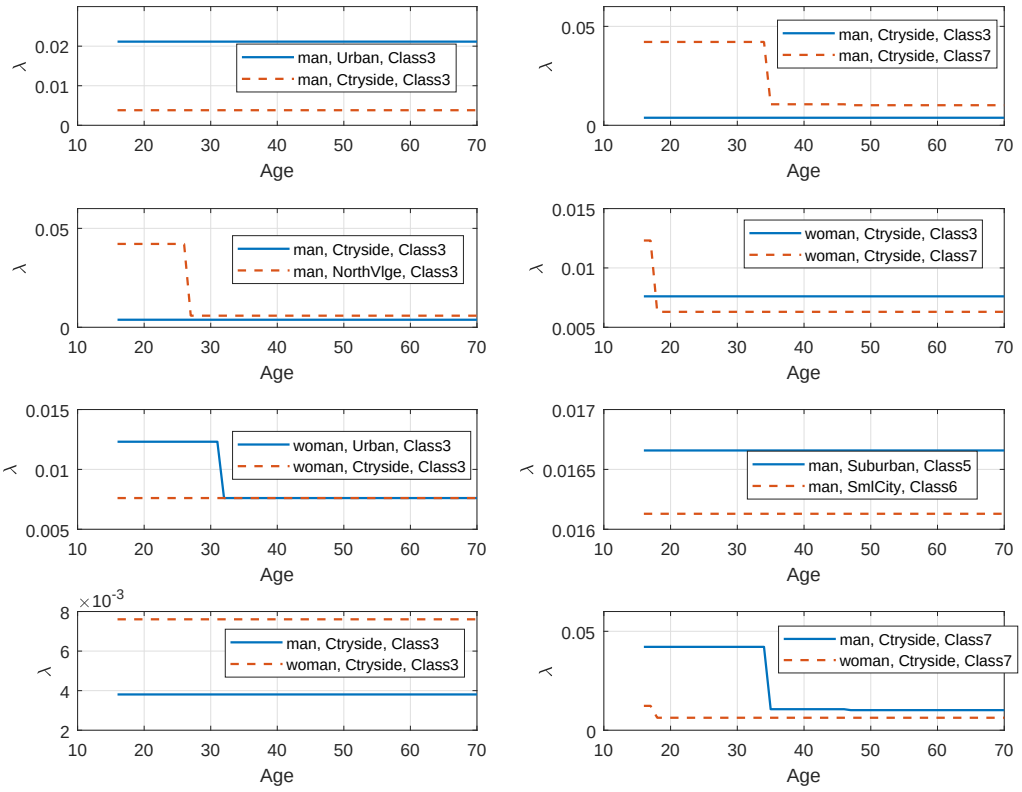

Figure 7: Regressed claims frequencies for different profiles of insured.

6 Conclusions

We demonstrate the usefulness of self-organizing maps for analysing insurance data. Self-organizing maps are unsupervised algorithms and do not need any a priori knowledge about the dependence structure between explanatory variables. Therefore they are useful for detecting collinearity and to design modalities of covariates feeding a multilayer perceptron or a generalized linear model. We present four variants of the Kohonen's algorithm and compare them to modified k-means procedures. The first one allows to pool policyholders based on quantitative explanatory variables. The second algorithm allows regressing the claims frequency on quantitative variables. Due to the low frequency of claims, we propose a Bayesian model to avoid a null probability of claims occurrence for some categories of insureds. With this model, the deviance, which is a traditional measure of the goodness of fit, is not necessary reduced when segmentation increases due to credibility weights. The k-means algorithm produces similar results to SOMs. The third algorithm studies

the dependence between categorical variables. It uses a χ^2 distance similar to the one used in a multiple correspondence analysis. This efficient procedure detects dominant variables in the portfolio. The k-means algorithm clearly under-performs compared to SOMs. The fourth procedure both segments policyholders and regresses claims frequency on quantitative and categorical variables. This approach helps to understand the geography of risks by assigning to a dominant profile of insureds to each neuron of the SOM. SOM seems to perform better than the k-means algorithm but is more computationally time consuming.

References

[1] Arthur D., and Vassilvitskii S. 2007. K-means++: The Advantages of Careful Seeding. SODA '07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027-1035.

[2] Benzécri, J.P. 1973. L'analyse des données, T2, l'analyse des correspondances, Dunod, Paris.

[3] Brockett P. , Cooper W. , Golden L., Pitaktong U 1994. A Neural Network Method for Obtaining an Early Warning of Insurer Insolvency. The Journal of Risk and Insurance, Vol 61(3), pp. 402-424.

[4] Brockett P., Xia X., Derrig. R. 1998. Using Kohonen's Self-Organizing Feature Map to Uncover Automobile Bodily Injury Claims Fraud. The Journal of Risk and Insurance, Vol 65(2), pp. 245-274.

[5] Brockett P. , Golden L., Jang J., Yang C. 2006. A Comparison of Neural Network, Statistical Methods, and Variable Choice for Life Insurers' Financial Distress Prediction. The Journal of Risk and Insurance, Vol 73(3), pp 397-419.

[6] Burt, C. 1950. The factorial analysis of qualitative data, British Journal of Psychology, Vol 3, 166-185.

[7] Cottrell M., Ibbou S., Letrémy P. 2004. SOM-based algorithms for qualitative variables. Neural networks, Vol 17, pp 1149-1167.

[8] Greenacre, M.J. 1984. Theory and Applications of Correspondence Analysis, London, Academic Press.

[9] Hainaut D., 2017. A neural network analyzer for mortality forecast. Forthcoming in the ASTIN Bulletin.

[10] Huysmans J., Baesens B., Vanthienen J., Van Gestel T. 2006. Failure prediction with self organizing maps. Expert Systems with Applications, Vol 30 (3), pp 479-487.

[11] Hertz, J., Krogh, A., Palmer, R.G. 1991. Introduction to the Theory of Neural Computation, Reading, MA : Addison-Wesley.

[12] Kohonen T. 1982. Self-Organized Formation of Topologically Correct Feature Maps. Biological Cybernetics. Vol 43 (1): 59–69.

[13] Lebart, L., Morineau, A., Warwick, K.M. 1984. Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices, Wiley.

[14] Lloyd, S. P., 1957. Least square quantization in PCM. Bell Telephone Laboratories Paper. Published in journal much later: Lloyd., S. P., 1982. Least squares quantization in PCM. IEEE Transactions on Information Theory. Vol 28 (2), pp 129–137.

[15] Ogunnaike R.M. Si D. 2017. Prediction of Insurance Claim Severity Loss Using Regression Models. Proceedings of the 13th Conference on Machine Learning and Data Mining. Springer eds. pp 233-247.

[16] Ohlsson E., Johansson B. 2010. Non-Life Insurance Pricing with Generalized Linear Models. Springer.

[17] Rosenblatt F. 1958. The perceptron: a probabilistic model, for information storage and organization in the brain. Psychological Review, Vol 65(6), pp 386-408.

[18] Sakthivel K. M. and Rajitha C.S. 2017. A Comparative Study of Zero-inflated, Hurdle Models with Artificial Neural Network in Claim Count Modeling. International Journal of Statistics and Systems. Vol 12(2), pp. 265-276.

[19] Wütrich M., Buser C. 2017. "Data Analytics for Non-Life Insurance Pricing". Lectures notes, available on https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2870308

[20] Viaene, S., Derrig, R. A., Baesens, B. and Dedene, G. 2002. A Comparison of State-of-the-Art Classification Techniques for Expert Automobile Insurance Claim Fraud Detection. The Journal of Risk and Insurance, Vol 69, pp. 373–421.

Acknowledgment

I gratefully acknowledges the BNP Cardiff Chair "Data Analytics and Models for Insurance" for its financial support. I also thank Michel Denuit from the UCL for his constructive advices.