INSURANCE ANALYTICS WITH CLUSTERING TECHNIQUES

Charlotte Jamotton, Donatien Hainaut, Thomas Hames

LIDAM Discussion Paper ISBA 2023 / 02

UCLouvain

ISBA | LIDAM Louvain Institute of Data Analysis and Modeling in economics and statistics

ISBA

Voie du Roman Pays 20 - L1.04.01

B-1348 Louvain-la-Neuve

Email : lidam-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/isba/publication.html

Insurance analytics with clustering techniques

Charlotte Jamotton [*], *UCLouvain*
Donatien Hainaut[†], *UCLouvain*
Thomas Hames[‡], *Detralytics*

12th January 2023

Abstract

The k-means algorithm and its variants are popular clustering techniques. Their purpose is to uncover group structures in a dataset. In actuarial applications, these partitioning methods detect clusters of policies with similar features and allow one to draw up a map of dominant risks. The main challenge lies in defining a distance between two observations exclusively characterised by categorical variables. This research paper starts with a review of the k-means algorithm and develops an extension based on Burt's framework to manage categorical rating factors. We then focus on a mini-batch version that keeps computation time under control when analysing a large-scale dataset. We next broaden the scope of application of the fuzzy k-means to fully categorised datasets. Lastly, we conclude with a thorough introduction to spectral clustering and work around the dimensionality issue by reducing the size of the initial dataset with k-means.

Keywords : clustering analysis, unsupervised learning, k-means, spectral clustering.

1 Introduction

Cluster analysis is one of the popular techniques within statistical data analysis and machine learning, helping to uncover group structures in data. Objects are grouped so that the resulting clusters are as heterogeneous as possible between each other while also being as homogeneous as possible with regard to the observations classified within them. In actuarial applications, clustering methods can detect dominant groups of policies, and the analysis of their claims allows one to draw a posteriori a map of insured risks. The resulting clusters can be further used to construct unsupervised pricing grids.

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

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

[‡]Postal address: Rue Belliard 2 - B-1040 Brussels (Belgium). E-mail to: t.hames(at)detralytics.be

The starting point of our work is the k-means algorithm, one of the most popular unsupervised learning algorithms (see, e.g., MacQueen (1967), Hastie et al. (2009) and Kaufman and Rousseeuw (2009)). Known as a centroid-based partitioning method, it segregates observations into a pre-specified number of clusters by optimising a measure of similarity. There exist several extensions of the k-means algorithm, including a mini-batch version to handle large-scale datasets. For a thorough overview, we refer to Jain (2010).

Despite their popularity in other fields such as image processing, clustering techniques are still underexploited in actuarial science, and the literature is scarce. Nevertheless, Williams and Huang (1997) may be referenced for their use of the k-means to identify policyholders with a high claims ratio in a portfolio of motor vehicle insurance. Hainaut (2019) further compared k-means to self-organizing maps (SOM) to discriminate motorcycle insurance policies based on the analysis of joint frequencies of categorical variables. We can also mention Hsu et al. (1999), who present SOM in a framework through which they perform change of representation in knowledge discovery problems using insurance policy data.

Clustering techniques are not, however, limited to the aforementioned centroid-based methods. In 1999, Weiss had already stressed the attractiveness of methods based on the eigenvectors of the affinity matrix in the context of segmentation. The stability of the simple eigendecomposition algorithms is often put forward in favour of spectral decomposition. Interested readers may refer for details to Shi and Malik (2000), Ng, Jordan and Weiss (2001) and Belkin and Niyogi (2001). Von Luxburg (2007) also sets up a concise step-by-step tutorial on the graph theory used by spectral clustering. Despite the overwhelming amount of literature on spectral clustering, no application to an insurance portfolio has hitherto been carried out.

The main reason behind the under-exploration of clustering techniques in actuarial science is the exclusive management of quantitative variables in their standard form by partitioning methods. In such cases, algorithms rely on the Euclidean distance between data points to measure their similarity. Some extensions, aiming to take qualitative variables into account already exist. For instance, Zhexue Huang (1998) extends the k-means algorithm to handle datasets with observations characterised both by quantitative and categorical variables. To this end, Zhexue Huang (1998) measures the distance between two observations by relying on a mixture of the Euclidean distance between quantitative features and Hamming's distance between (binary coded) categorical features. Similarly, Mbuga and Tortora (2021) recently extended spectral clustering to heterogeneous datasets, with both continuous and categorical features, by defining a global dissimilarity measure computed using a weighted sum. In their paper on feature transformation, Wei, Chow and Chan (2015) outline the challenges of developing a dissimilarity measure as a linear combination of the Euclidean and Hamming distances to handle mixed-type datasets. Xiong et al. (2012) further proposed a novel divisive hierarchical clustering algorithm for categorical objects. Starting with an initialization based on multiple correspondence analysis, they exploit the Chi-square distance between a single object and a set of objects to define the objective function of their clustering algorithm.

The contributions of this research paper are multiple. The main premise of this work is the data processing through Burt's space. It allows an investigation into different methods of data partitioning for actuarial applications. In Section 1, we introduce the centroid-based k-means algorithm and review its k-means++ heuristic. We next extend it to fully categorised datasets, based on Burt's distance. Section 2 reviews a mini-batch version to perform clustering on large-scale datasets with a limited loss of accuracy. In Section 3, we broaden the scope of application of the fuzzy k-means by means of our Burt framework. We further test its efficiency as a method in which policies can belong to multiple clusters. We end this work in Section 5 with a thorough review of spectral clustering, as an unprecedented means to identify non-convex clusters in an insurance portfolio. The need for the eigendecomposition makes spectral clustering more computationally expensive than standard k-means for large-scale datasets. Reducing the size of the initial dataset with the k-means algorithm and applying spectral clustering to the resulting centroids shows itself to be an effective solution to work around this dimensionality issue.

2 The k-means algorithm and Burt's distance

Let us consider a set of n numeric objects $X = \{\boldsymbol{x}_1, ...\boldsymbol{x}_n\}$, $\boldsymbol{x}_i \in \mathbb{R}^p$ and an integer number $K \leq n$. The *k-means* algorithm searches for a partition of the data set X into a pre-specified number K of disjoint clusters such that the within groups sum of squared errors (WGSS) or intraclass inertia in Eq.(2) is minimised. The k-means clustering method is based on the concept of a centroid, which may be interpreted as the centre of gravity of a cluster of objects. As it is defined as the geometric centre of the corresponding data cluster, a centroid is not necessarily one of the data points. The coordinates of the u^{th} centroid are contained in a vector $\boldsymbol{c}_u = (c_1^u, \ldots, c_p^u)$ for $u = 1, ..., K$. We define the clusters or groups of data S_u for $u = 1, ..., K$ for a given distance $d(.,.)$ and a set of K centroids as follows:

$$S_u = \{\boldsymbol{x}_i \ : \ d(\boldsymbol{x}_i, \boldsymbol{c}_u) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_v) \ \forall v \in \{1, ..., K\}\}\,. \tag{1}$$

Here, the dissimilarity measure

$$d(\boldsymbol{x}_i, \boldsymbol{c}_u) = \sum_{j=1}^{p} \left(x_{i,j} - c_j^u\right)^2 u = 1, ..., K$$

is the squared Euclidean distance. The centre 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 centre of gravity of the full dataset is denoted by $\boldsymbol{g} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$. We define the inertia I_u of a cluster S_u as the sum of the distances of all the points within a cluster from the centroid of that cluster:

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

The intraclass inertia I_a is the sum of clusters' inertia, weighted by their size:

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

whereas the interclass inertia I_c is the inertia of the cloud of centres of gravity:

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

According to the König-Huygens theorem, the global inertia is the sum of the intraclass and interclass inertia:

$$\begin{aligned} I_X &= I_c + I_a \\ &= \frac{1}{n} \sum_{i=1}^{n} d\left(\boldsymbol{x}_i, \boldsymbol{g}\right)^2 \end{aligned}$$

In order to have homogeneous clusters on average (compact clusters), an usual criterion of classification consists in identifying a partition of X that minimises the intraclass inertia I_a. This amounts to finding the partition that maximises the interclass inertia, I_c, which ensures the heterogeneity between clusters. Optimizing such objective functions is known to be computationally difficult (NP-hard). However, there exist efficient heuristic procedures that quickly converge to a local optimum with respect to interclass inertia. The most common method uses an iterative refinement technique called k-means, which is detailed in Algorithm 2.

Given an initial set of K random centroids $\boldsymbol{c}_1(0),\ldots,\boldsymbol{c}_K(0)$, we partition $\{S_1(0), \ldots, S_K(0)\}$ the dataset according to the rule in equation (1). As an alternative to this random centroids initialization, the *k-means++* algorithm of Vassilvitskii and Arthur (2006) hinges on a heuristic to find the centroid seeds $\boldsymbol{c}_1(0),\ldots,\boldsymbol{c}_K(0)$ and construct the initial partition of the dataset. This alternative procedure to initialise the k-means heuristic is detailed in Algorithm 1. When compared to the random centroids initialization, it improves the running time of the algorithm, and the quality of the final solution. Keep in mind that the k-means clustering is based on a heuristic. The resulting partition is therefore not guaranteed to be a global solution, and changes in the seed value may affect the outcomes.

Regardless of the procedure chosen for the arbitrary centroids initialization, the resulting partition of the dataset is a set of convex polyhedrons delimited by median hyperplanes of centroids. Once the K centroids are initialised, we proceed by alternating between two steps. The assignment step of the e^{th} iteration associates each observation $\boldsymbol{x}_i$ with the cluster $S_u(e)$ whose centroid $\boldsymbol{c}_u(e)$ is the nearest. In the update step, the K centroids

$(\boldsymbol{c}_u(e+1))_{u=1:K}$ are replaced by the K new cluster means $(\boldsymbol{g}_u(e))_{u=1:K}$. At each iteration, we can prove that the intraclass inertia is reduced. We iterate until convergence, which is when the assignments no longer change.

Algorithm 1 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 dataset, 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

Algorithm 2 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 data point $\boldsymbol{x}_i$ to the cluster $S_u(e), u \in \{1, ..., K\}$ whose centre $\boldsymbol{c}_u(e)$ is closest

$$S_u(e) = \{\boldsymbol{x}_i \ : \ d(\boldsymbol{x}_i, \boldsymbol{c}_u(e)) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_v(e)) \ \forall v \in \{1, ..., K\}\}.$$

End loop on data set, 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

To illustrate this section, we apply the k-means algorithm to data from the Swedish insurance company *Wasa* in 1999. The data set is available on the companion website of the book by Ohlsson and Johansson (2010) and contains information about motorcycle insurance 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 vehicle's class. The vehicle class is determined by the ratio of power in KW $\times 100$ / vehicle weight in kg + 75, rounded to the nearest integer. The database also reports for each policy the number of claims, the total claim costs, and the contract period. After removing contracts with null duration, the database counts $n = 62\,436$ insurance policies. Table 1 summarises the information provided by categorical variables.

Rating factors	**Class**	**Class description**
Gender	M	Male (ma)
	K	Female (kvinnor)
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	Northern towns
	6	Northern 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 for motorcycle insurance. Source: Ohlsson and Johansson (2010).

The p numerical variables are stored in a vector $\boldsymbol{x}_{j=1,...,n} \in \mathbb{R}^p$, and the q categorical variables have m_k binary modalities for $k = 1, ..., q$. The total number of modalities is the sum of m_k: $m = \sum_{k=1}^{q} m_k$. In further developments, we enumerate modalities from 1 to m. The information about the portfolio may be summarised by a $n \times m$ super-indicator matrix $D = (d_{i,j})_{i=1...n, j=1...m}$. If the i^{th} policy has the j^{th} modality then $d_{i,j} = 1$, otherwise $d_{i,j} = 0$.

For example, assume policies are exclusively described by the policyholder's gender (M=male or F=Female) and by his education level (H=high school, C=college or U=university). The number of variables and modalities are $q = 2$, $m_1 = 2$ and $m_2 = 3$ respectively. If the first and second policyholders are respectively a man with an undergraduate degree and a

woman with a graduate degree, the two first lines of the matrix D have the structure of the disjunctive Table 2.

	Gender		Degree		
Policy	M	F	H	C	U
1	1	0	0	1	0
2	0	1	0	0	1

Table 2: Example of a disjunctive table with $q = 2$ variables and $m_1 = 2$, $m_2 = 3$ modalities, respectively.

In this paper, we focus on features encoded as categorical variables. In such case, the Euclidean distance is not appropriate anymore. When dealing with one-hot encoded categorical features, Hamming's distance is usually the first substitute that comes to mind. It measures the dissimilarity between features as the number of different bit positions in two bit strings (the u^{th} centroid and the i^{th} policy bit strings). The disappointing results achieved with Hamming's distance lead us to another measure based on the weighted Burt matrix.

2.1 Burt's distance

Hamming's distance is a simple measure of discordance between observations. It therefore fails to discriminate between observations that are "far" from those that are "close" to each other. This issue is addressed by Burt's distance, which is based on the study of joint frequencies in modalities. In order to study the dependence between modalities, the numbers $n_{i,j}$ of policies sharing modalities i and j, for $i, j = 1, ..., m$ is contained in a contingency table, known as the $m \times m$ Burt matrix $\boldsymbol{B} = (n_{i,j})_{i,j=1,...,m}$. The Burt matrix is directly related to the disjunctive table D through the following mathematical relationship:

$$\boldsymbol{B} = \boldsymbol{D}^\top \boldsymbol{D}\,.$$

This symmetric matrix is composed of $q \times q$ blocks $\boldsymbol{B}_{k,k'}$ for $k, k' = 1, ..., q$. A block $\boldsymbol{B}_{k,k'}$ is the contingency table that crosses the variables k and k'. Table 3 displays the Burt matrix for matrix $\boldsymbol{D}$ presented in Table 2. By construction, the sum of elements of a block $\boldsymbol{B}_{k,k'}$ is equal to the total number of policies, n. The sum of $n_{i,j}$ for row i is equal to:

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

Because the Burt matrix is symmetric, we can infer that

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

Furthermore, blocks $\boldsymbol{B}_{k,k}$ for $k = 1, ..., q$ are diagonal matrices, whose diagonal entries are the numbers of policies that respectively present the modalities $1, ..., m_k$, for the k^{th}

variable. In our example, we have $n_{1,1}+n_{2,2}=n$ and $n_{3,3}+n_{4,4}+n_{5,5}=n$. Here, $n_{1,1}$ and $n_{2,2}$ count the total number of men and women in the portfolio, whereas $n_{3,3}$, $n_{4,4}$ and $n_{5,5}$ count the number of policyholders with, respectively, a high school, college, or university degree.

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

Table 3: Burt matrix for the disjunctive Table 2.

In the same manner as Hainaut (2019), we define the chi-square distance between rows i and i' of the Burt matrix as follows:

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

Intuitively, the distance between two modalities is measured by the sum of weighted gaps between joint frequencies with respect to all modalities. Similarly, the chi-square distance between columns j and j' of the Burt matrix is defined by

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

We recall that k-means is a centroid-based clustering algorithm. Bregman divergences are the only distortion functions for which such centroid-based clustering schemes are possible. The simplest divergence is known to be the squared Euclidean distance. As we prefer to evaluate distances with the Euclidean distance, the elements of the Burt matrix $n_{i,j}$ are replaced by weighted values $n_{i,j}^{W}$:

$$n_{i,j}^{W} \quad := \quad \frac{n_{i,j}}{\sqrt{n_{i,.}\, n_{.,j}}} \qquad i,j = 1,...,m\,. \tag{3}$$

Given that $n_{i,.} = q\, n_{i,i}$ and $n_{.,j} = q\, n_{j,j}$, we have that

$$n_{i,j}^{W} \quad := \quad \frac{n_{i,j}}{q\sqrt{n_{i,i}\, n_{j,j}}} \qquad i,j = 1,...,m\,. \tag{4}$$

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

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

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

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

The j^{th} modality then corresponds to the j^{th} line of $\boldsymbol{B^W}$, a vector in $\mathbb{R}^m$. The centre of gravity $\boldsymbol{D}_{i,.}\boldsymbol{B^W}/q$ of points with coordinates stored in the corresponding lines of the weighted Burt matrix can then be used to identify the i^{th} policy with multiple modalities. Since a policy i may fall into a single one of the m_k categories of the k^{th} variable, a policy characterised by two features is also defined by a subset of $q = 2$ modalities. We represent each policy in $\mathbb{R}^3$ as the midpoint between the corresponding lines of $\boldsymbol{B^W}$. The Burt's distance between the i^{th} and j^{th} policies is then given by:

$$d(i, j) \quad = \quad \left\| \boldsymbol{D}_{i,.}\boldsymbol{B^W}/q - \boldsymbol{D}_{j,.}\boldsymbol{B^W}/q \right\|_2 \tag{5}$$

This is illustrated in Figure 1 in the case of three modalities.

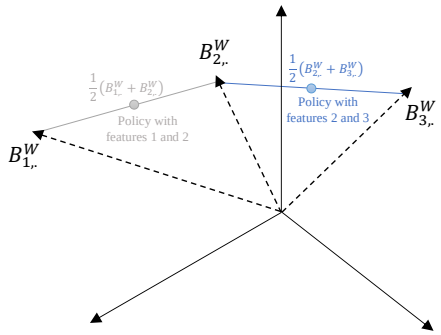

Figure 1: Illustration of two policies in the Burt's space with three modalities.

The second step of Algorithm 2 must thence be adapted. The new centroids $\boldsymbol{c}_u(e+1) = \boldsymbol{c}_u^m(e+1)$ of $S_u(e)$ are defined by a m-vector, $(\boldsymbol{c}_u^m(e+1))_{u=1:K}$, containing the centre of gravity of the categorical variables in cluster u:

$$\boldsymbol{c}_u^m(e+1) = \frac{1}{|S_u(e)|} \sum_{\boldsymbol{x}_i \in S_u(e)} \boldsymbol{D}_{i,.}\boldsymbol{B^W}/q\,.$$

We implement this procedure on the Wasa insurance dataset fully converted into categorical variables. We first convert the variables "driver's age" and "vehicle age" into categorical variables with respectively 6 and 5 modalities, [16, 28), [28, 30), [30, 34), [34, 54), [54, 60), [60, 99) and [0, 2), [2, 4), [4, 6), [6, 12), [12, 100). We next compute the matrix of $\boldsymbol{x}_i = \boldsymbol{D}_{i,.}\boldsymbol{B^W}/q$ for $i = 1$ to n and apply the standard k-means to the distance exclusively evaluated with the weighted Burt's table.

Table 4 reports the claim frequency as well as the dominant modalities associated with each cluster. The dominant modality $j_k^{dominant}$ is identified as either the dominant

representative in the u^{th} cluster or as the one that minimises the distance measured by the ℓ^2-norm between the centroid $(c_u)_{u=1:K}$ and the line of B^W corresponding to the j^{th} modality of the k^{th} variable:

$$j_k^{dominant} = \arg \min_{v \in \{1,\dots,m_k\}} \left\| c_u - B^W_{\sum_{d=1}^{k-1} m_d + v,.} \right\|_2$$

A quick analysis reveals the most and least risky driver's profiles. The riskiest category is young male drivers of vehicles of less than 4 years. The lowest claim frequency is characterised by older vehicles (12+). The under-representation of female drivers in the portfolio is also reflected in the dominant gender (mostly men) associated to each cluster.

Cluster	Gender	Zone	Class	Owner's Age	Vehicle Age	Frequency (%)
1	M	4	3	[34, 54)	[12, 100)	0.2769
2	M	4	6	[16, 28)	[6, 12)	1.9901
3	K	4	3	[34, 54)	[6, 12)	1.4439
4	M	3	4	[34, 54)	[6, 12)	1.3062
5	M	3	6	[16, 28)	[6, 12)	3.2422
6	M	4	3	[30, 34)	[12, 100)	1.4061
7	K	4	3	[34, 54)	[12, 100)	0.4425
8	M	2	5	[34, 54)	[12, 100)	0.3662
9	M	3	4	[34, 54)	[12, 100)	0.446
10	K	4	3	[16, 28)	[12, 100)	0.7167
11	M	4	3	[60, 99)	[12, 100)	0.3097
12	M	3	3	[16, 28)	[2, 4)	4.3287
13	M	4	3	[54, 60)	[12, 100)	0.1662
14	M	1	3	[34, 54)	[12, 100)	0.8051
15	M	4	4	[34, 54)	[4, 6)	0.9937
16	M	4	5	[16, 28)	[12, 100)	1.7233
17	M	4	3	[34, 54)	[6, 12)	0.4281
18	M	4	4	[34, 54)	[2, 4)	1.3045

Table 4: K-means algorithm on Burt's distance. Dominant features and average claim frequencies per cluster.

If we use the claim frequency as a predictor, $\widehat{\lambda}_i$, we can estimate the goodness of fit of the partition with the Poisson deviance. The deviance is the difference between the log-likelihood of the saturated model and that of the partitioned model. If N_i and ν_i are respectively the number of claims and the duration (exposure) of the i^{th} contract, the deviance is defined as:

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

The Deviance, AIC, and BIC are reported in Table 5. Both the AIC and BIC are computed with a number of degrees of freedom equal to 18×27 (# of clusters × # of modalities). In comparison, the deviance of a GLM model fitted to the same dataset is equal to 5732.

Goodness of fit	
Deviance	6 082.025
AIC	8 402.62
BIC	12 796.98

Table 5: Statistics of goodness of fit obtained by partitioning the dataset into 18 clusters with the k-means algorithm based on Burt's distance.

Figure 2 displays the 18 clusters in a two-dimensional space with the owner's and vehicle's ages on the axes.

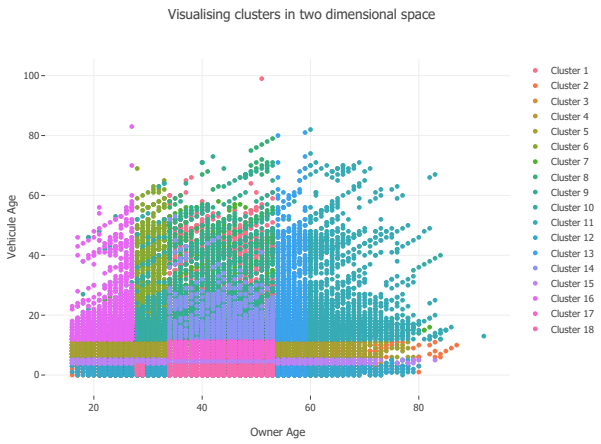

Figure 2: Illustration of the partitioning of a dataset into 18 clusters with the k-means algorithm based on Burt's distance.

As aforementioned, both the random and k-means++ procedures used to initialise the k-means heuristic require the specification of an arbitrary number of clusters K. The criteria for choosing an optimal number of clusters is usually based on the marginal gain of intra-class inertia or the marginal reduction of deviance. The marginal gain (reduction) of inertia (deviance) is limited above a certain number of clusters. Figure 3 illustrates the inertia and deviance for various levels of segmentation of the Wasa portfolio, according to variables "Owner's age", "Vehicle age", "Gender", "Zone", and "Class".

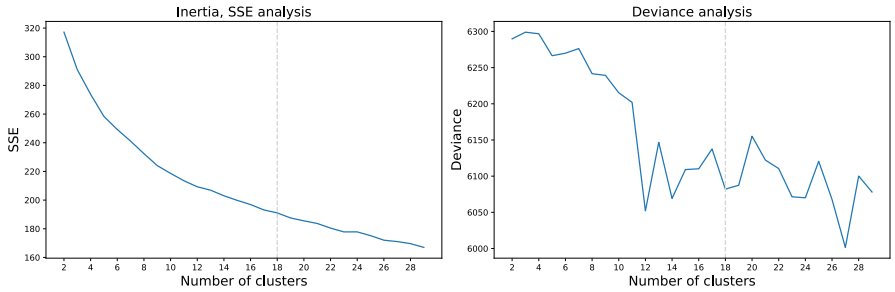

Figure 3: Left plot: evolution of the total intra-class inertia. Right plot: evolution of the Deviance.

3 Mini-batch k-means

The k-means algorithm stores the entire dataset in main memory. When dealing with large amounts of data, this characteristic can turn into a major computation time constraint. The mini-batch k-means algorithm is a popular alternative to reduce the temporal and spatial costs. The underlying idea hinges on saving memory space by using small random batches of data with a fixed size. At each iteration, a new random sample from the dataset is used to update the clusters, taking care of deprecating previous coordinates according to a learning speed. This step is repeated until convergence. The details of this approach for categorical variables combined with the Burt's distance are set forth in Algorithm 3.

Empirical results in the literature suggest a substantial saving in computational time at the expense of partition quality. Figure 4 compares the standard k-means algorithm with the mini-batch's relative savings in computational time with respect to the number of clusters.

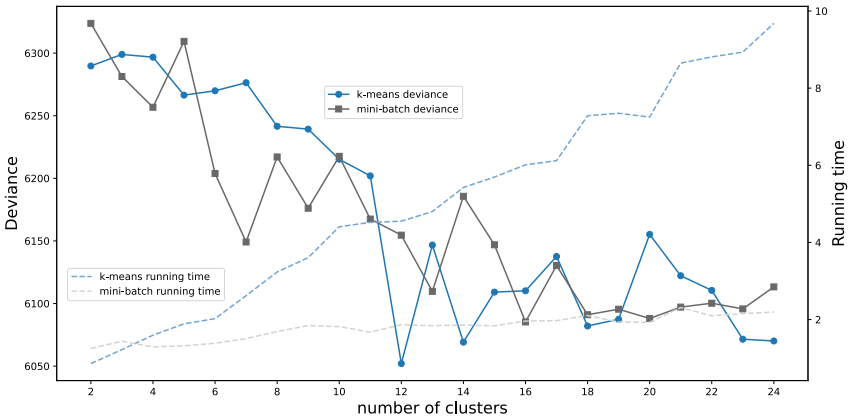

Figure 4: Partition quality (measured by the deviance in solid lines) and running time (dotted lines) with respect to the number of clusters.

Algorithm 3 Mini-batch k-means algorithm

Initialization:

Randomly set up initial positions of k centroids

Initialize clusters $S_1 = ... = S_K = \emptyset$

Main procedure:

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

Random sampling of the batch dataset M of size b

Initialize sample clusters $S_1^{new} = ... = S_K^{new} = \emptyset$

Assignment step:

For $i = 1$ to b

1) Assign i^{th} policy to cluster S_u^{new} where

$$S_u^{new} = \{u \,:\, d(i\,, \boldsymbol{c}_u(e)) \leq d(i\,, \boldsymbol{c}_v(e))\ \forall v \in \{1, ..., K\}\}\,.$$

End loop on batch dataset, i.

Update step:

For $u = 1$ to K

2) Calculate the centroids of the batch assigned to S_u^{new}:

$$\boldsymbol{c}_u^{m,new} = \frac{1}{|S_u^{new}|} \sum_{i \in S_u^{new}} \boldsymbol{D}_{i,.} \boldsymbol{B}^{\boldsymbol{W}} / q\,.$$

3) Let $\eta_u(e) = \frac{|S_u^{new}|}{|S_u^{n}| + |S_u^{new}|}$. Centroids $\boldsymbol{c}_u^m(e+1)$ of S_u are:

$$\boldsymbol{c}_u^m(e+1). = (1 - \eta_u(e))\, \boldsymbol{c}_u^m(e) + \eta_u(e) \boldsymbol{c}_u^{m\,new}\,.$$

End loop on centroids, u.

End loop on epochs e

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

We run the mini-batch algorithm on the Wasa dataset with Burt's distance (5) and batches of 5000 policies. Table 7 provides the results for 18 centroids. The mini-batch k-means is faster and gives approximately the same results. We retrieve all the categories discovered in Sub-section 2.1. The cluster with the second highest frequency is also characterised by one of the highest ratio powers (class 6). The deviance, reported in Table 6 is slightly higher than that of a segmentation based on the k-means algorithm. Keep in mind that the small loss in cluster quality may be the product of a greater presence of hazards in the mini-batch algorithm.

In our case study, the gain in computation time with respect to k-means is limited (both algorithms run in a few seconds). To observe significant gains in computation time, the mini-batch k-means should be tested on a larger dataset than the Wasa one.

Goodness of fit	
Deviance	6 091.04
AIC	8 411.31
BIC	12 805.97

Table 6: Statistics of goodness of fit obtained by partitioning the dataset into 18 clusters with the mini-batch k-means.

Cluster	Gender	Zone	Class	Owner's Age	Vehicle Age	Frequency (%)
1	M	4	3	[34, 54)	[12, 100)	0.2769
2	M	3	3	[16, 28)	[2, 4)	4.5118
3	M	3	6	[16, 28)	[6, 12)	3.4312
4	M	4	4	[54, 60)	[6, 12)	1.3168
5	M	2	3	[34, 54)	[12, 100)	0.3991
6	M	4	1	[60, 99)	[12, 100)	0.4124
7	K	4	3	[34, 54)	[12, 100)	0.3834
8	M	3	3	[34, 54)	[6, 12)	0.7141
9	M	2	5	[54, 60)	[12, 100)	1.2385
10	M	3	4	[34, 54)	[12, 100)	0.4401
11	K	4	6	[34, 54)	[6, 12)	1.3198
12	M	2	4	[16, 28)	[6, 12)	3.1343
13	K	2	3	[16, 28)	[12, 100)	1.2797
14	M	2	3	[34, 54)	[2, 4)	1.5865
15	M	1	3	[34, 54)	[6, 12)	1.8939
16	M	4	3	[34, 54)	[6, 12)	0.5118
17	M	4	3	[54, 60)	[12, 100)	0.2933
18	M	4	6	[16, 28)	[12, 100)	2.1356

Table 7: Coordinates of centroids and average claim frequencies for 18 clusters, mini-batch k-means.

4 Fuzzy k-means

The k-means algorithm, also known as 'hard clustering', attempts to partition the data set X into K distinct clusters with respect to a cost function. Each data point is assigned to the cluster whose centre is the nearest and may only belong to one cluster. Fuzzy k-means, also referred to as 'soft clustering', distinguishes itself by allowing data points to potentially belong to more than one cluster S_u for $u = 1, ..., K$. Given a finite set of data, the algorithm returns a list of K cluster centres $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_K\}$ and a partition matrix W of "membership" $w_{i,j}$ for $i = 1, ..., n$ and $j = 1, ..., K$. The $w_{i,j}$ gives the degree to which the i^{th} policy belongs to cluster S_j. Much like the k-means algorithm, the fuzzy k-means

algorithm aims to minimise the intra-cluster variance:

$$\arg\min \sum_{i=1}^{n} \sum_{j=1}^{K} (w_{i,j})^m \, d(i, \boldsymbol{c}_j)$$

where

$$w_{i,j} = \frac{1}{\sum_{u=1}^{K} \left(\frac{d(i,\boldsymbol{c}_j)}{d(i,\boldsymbol{c}_u)} \right)^{\frac{2}{m-1}}}.$$

The hyper-parameter $m \in \mathbb{R}^+$ with $m \geq 1$ is called the fuzzifier. It determines the level of cluster fuzziness. A large m results in smaller membership grades $w_{i,j}$, and hence fuzzier clusters. In the limit $m = 1$, the memberships, $w_{i,j}$, converge to 0 or 1, which amounts to crisp partitioning.

The arbitrary centroid initialization and the pre-specified number of clusters both retain their significant influence over the final partition, which is not guaranteed to be a global optimum. The fuzzy k-means is detailed in Algorithm 4 for categorical variables combined with Burt's distance.

Algorithm 4 Fuzzy 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) Calculate the probability that the i^{th} policy is in cluster $S_j(e)$, $j \in \{1, ..., K\}$

$$w_{i,j} = \frac{1}{\sum_{u=1}^{k} \left(\frac{d(i\,,\boldsymbol{c}_j)}{d(i\,,\boldsymbol{c}_u)} \right)^{\frac{2}{m-1}}}.$$

End loop on data set, i.

Update step:

For $u = 1$ to K

2) Update centroids $\boldsymbol{c}_u(e+1) = \boldsymbol{c}_u^m(e+1)$ of $S_u(e)$:

$$\boldsymbol{c}_u^m(e+1) = \frac{\sum_{i=1}^{n} w_{i,u}(e)^m \boldsymbol{D}_{i,\cdot} \boldsymbol{B}^{\boldsymbol{W}}/q}{\sum_{i=1}^{n} w_{i,u}(e)^m}$$

End loop on centroids, u.

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

$$d^{total} = \sum_{u=1}^{K} \sum_{i=1}^{n} w_{i,u}(e)^m d(i\,, \boldsymbol{c}_u(e+1))\,.$$

End loop on epochs e

We apply fuzzy k-means to the Wasa insurance dataset fully converted into categorical variables. We convert the variables "driver's age" and "vehicle age" into categorical variables

as in the sub-section 2.1. We run the algorithm with a fuzziness parameter of $m = 1.10$. The policies are assigned to the most likely cluster (highest $w_{i,j}$ for $j = 1, ..., K$). Table 8 reports the most frequent features and the claim frequency in each cluster. We find similar profiles for the most and least risky drivers to those found in sub-section 2.1, but with a much higher claim frequency for the riskiest group.

Cluster	Gender	Zone	Class	Owner's Age	Vehicle Age	Frequency (%)
1	M	3	3	[34, 54)	[6, 12)	1.2458
2	M	3	3	[16, 28)	[12, 100)	0.7425
3	M	4	3	[34, 54)	[12, 100)	0.2769
4	M	4	3	[34, 54)	[2, 4)	0.875
5	M	2	3	[34, 54)	[12, 100)	0.467
6	K	2	3	[34, 54)	[6, 12)	1.2593
7	M	4	3	[60, 99)	[12, 100)	0.3647
8	M	2	4	[34, 54)	[2, 4)	1.6797
9	M	3	3	[34, 54)	[12, 100)	0.3353
10	M	3	6	[16, 28)	[6, 12)	3.5154
11	M	4	4	[34, 54)	[6, 12)	0.4699
12	M	1	3	[54, 60)	[2, 4)	0.9302
13	M	4	6	[16, 28)	[6, 12)	2.5934
14	K	4	3	[34, 54)	[12, 100)	0.5173
15	M	4	4	[54, 60)	[6, 12)	1.0525
16	M	2	4	[54, 60)	[2, 4)	2.8011
17	M	3	3	[16, 28)	[4, 6)	9.1737
18	M	2	3	[16, 28)	[12, 100)	1.6596

Table 8: Fuzzy clustering, dominant features, and average claim frequencies per cluster.

Figure 5 depicts the 18 clusters in the space delimited by tge owner and vehicle ages. The deviance (Table 9) is better than that of the (mini-batch) k-means. Notice that if we set $m = 2$, which is a standard level of fuzziness in the literature, several clusters are not assigned any policies, but some policies have a non-null probability of belonging to them.

Goodness of fit	
Deviance	6 025.51
AIC	8 346.08
BIC	12 740.44

Table 9: Fuzzy clustering. Statistics of goodness of fit obtained by partitioning the full categorical dataset into 18 clusters.

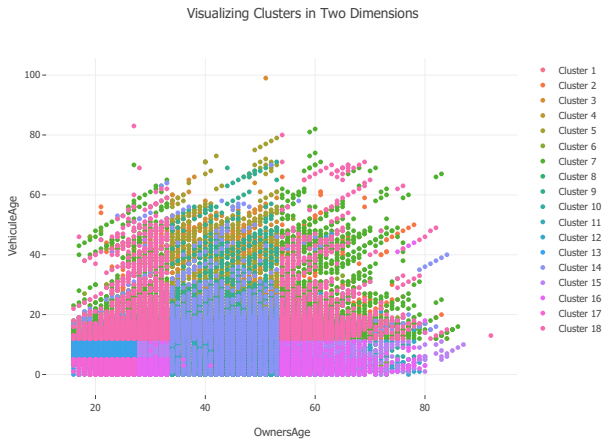

Figure 5: Illustration of the partitioning of a dataset with the fuzzy k-means algorithm based on Burt's distance.

5 Spectral clustering

Because the k-means algorithm relies on the Euclidean distance, every single iteration necessarily leads to a subdivision of the space into (hyper)spherical clusters. This makes the k-means clustering method non suitable for detecting clusters with non-convex geometrical representation. To illustrate its weak performance under this scenario, we apply the k-means algorithm to partition into two clusters the circular and moon-shaped data plotted in Figure 6. Unlike spectral clustering, (mini-batch) k-means fails to identify the inner and outer rings, as well as the upper and lower moons. This unsatisfactory clustering is the direct result of the natural clusters not qualifying as convex regions.

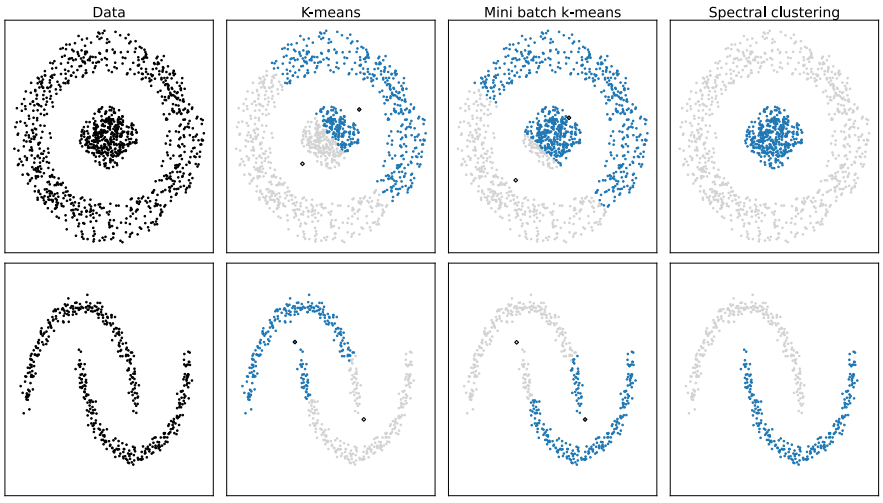

Figure 6: Illustration of the partition of non-convex data with the (mini-batch) k-means, and spectral algorithms. Each cluster is identified by a colour. The centroids are represented with a diamond shape.

By exploiting a deeper data geometry, spectral clustering (Shi and Malik 2000; Ng, Jordan and Weiss 2001; Belkin and Niyogi 2001) addresses the k-means lack of flexibility in cluster boundaries. While k-means attempts to associate each cluster with a hard-edged sphere, spectral clustering works by embedding the data in a different space derived from a graphical representation of the dataset. Applying the k-means algorithm to this representation bypasses the spherical limitation, and results in the intended non-convex cluster shapes.

The graph theory is the core component in the graph representation of the dataset. As illustrated in Figure 7, an undirected graph $G = (V, E, W)$ is defined by three elements:

1. a set of vertices $V = \{v_i\}_{i=1,...,n}$ where each vertex v_i represents one of the n data points in the dataset;

2. a set of edges $E = \{e_{i,j} : v_i \longleftrightarrow v_j\}$ whose entries are equal to 1 if two vertices v_i and v_j are connected by an undirected edge $e_{i,j}$, and 0 otherwise; and

3. a set of weights $W = \{w_{ij} : w_{ij} \neq 0 \text{ if } v_i \longleftrightarrow v_j\}$ that contains the similarity between two vertices linked by an edge.

E and W can both be represented as $n \times n$ matrices. The matrix W is often referred to as the weighted adjacency matrix A.

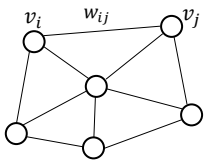

Figure 7: Vertices, edges and weights of a graph.

The dataset $X = \{\boldsymbol{x}_1, ...\boldsymbol{x}_n\}$, $\boldsymbol{x}_i \in \mathbb{R}^p$ is thence represented as a graph $G = (V, E, W)$ by first associating each data point x_i with a vertex v_i. A measure of similarity between two data points i and j is then defined. The most common is based on the Gaussian kernel

$$S(i, j) = \exp\left(-\frac{d(i, j)}{\alpha}\right),$$

where $\alpha \in \mathbb{R}^+$ is a tuning parameter. Two highly similar data points will be connected by an edge $e_{i,j}$ with a weight w_{ij} equal to their similarity measure. Data points with a low similarity are considered disconnected. There exist different ways to create the pairwise similarity graph representation:

a. The ϵ-neighbourhood graph: all the points for which the pairwise distances are smaller than ϵ are considered as connected. In practice, it means keeping all similarities smaller than ϵ and forcing all the others to 0.

b. The (mutual) k-nearest neighbour graph: the vertex v_i is connected with the vertex v_j if v_j is among the k-nearest neighbours of v_i. Since the neighbourhood relationship is not symmetric and the graph must be symmetric, we need to enforce the symmetry. To do so, we may simply choose to ignore the directions of the edges; we connect v_i and v_j if v_i is among the k-nearest neighbours of v_j **or** if v_j is among the k-nearest neighbours of v_i. The resulting graph is usually called the k-nearest neighbour graph. Another option would consist in connecting v_i and v_j if v_i is among the k-nearest neighbours of v_j **and** v_j is among the k-nearest neighbours of v_i. The resulting graph is usually called the *mutual* k-nearest neighbour graph.

c. The fully connected graph: we connect all the points available in the dataset.

Keep in mind that the arbitrary chosen rule will set the edge matrix's entries, and may influence the resulting clusters.

In order to work with the so obtained graph representation, we use a graph Fourier transform based on its Laplacian representation. The Laplacian representation of a graph G is derived from the difference between the degree matrix D and the adjacency matrix A :

$$L = D - A$$

Why is matrix L called graph Laplacian? We can define a function on a graph's vertices, $f : V \to \mathbb{R}$ such that $v_i \to f(v_i)$. Let us consider a discrete periodic function that takes N values, at times $1, 2, ..., N$. The loop on periods may be represented by a ring graph, as shown in Figure 8.

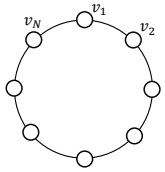

Figure 8: Ring representation of a period with N steps.

In this particular case, the matrix of edges and weights is:

$$A = E = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & \ddots & 0 \\ 0 & 1 & 0 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 & 1 \\ 1 & 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

Its entries indicate the absence or presence of a (weighted) edge between the vertices. Because none of the vertices have edges to themselves, its diagonal elements are 0. Because our graph is undirected, $E_{ij} = E_{ji}$.

In order to find the Laplacian representation, we subtract the adjacency matrix A set out above from the diagonal degree matrix D. The matrix D contains on its diagonal the degree of each vertex $D_{ii} = \sum_j w_{i,j}$. The degree of a vertex v_i represents the weighted number of edges connected to it, and is equal to the sum of the row i in the adjacency matrix.

If we denote by $\boldsymbol{f} = (f(v_j))_{j=1,\dots,N}$ the vector of values of $f(.)$ at vertices, the product $L\boldsymbol{f}$ corresponds precisely to the second finite difference derivative of the function $f(.)$.

$$L = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \ddots & 0 \\ 0 & -1 & 2 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

Both the adjacency and degree matrices are encased in the resulting Laplacian matrix; the diagonal entries are the degrees of the vertices, and the off-diagonal elements are the negative edge weights.

The partition of the graph, including the number of clusters, is then inferred from the eigenvalues and eigenvectors of the graph Laplacian matrix (spectral analysis). Since L is symmetric, we can rewrite Laplacian L as $U\Sigma U^\top$ where U is a matrix containing the eigenvectors and Σ a diagonal matrix containing the eigenvalues. As L is a positive semi-definite matrix, it can be easily shown that the smallest eigenvalue of L is always 0.

The analysis of the eigenvalues gives useful insights into the graph's structure. The eigenspace of 0 gives a way to partition the graph. For instance, if all the vertices of a graph are completely disconnected, all the eigenvalues are zero. As we add edges, some of the eigenvalues become non-null. The number of null eigenvalues corresponds to the number of groups of connected vertices in our graph. As an illustration, let us consider the graph plotted in Figure 9 that is made of K different groups of vertices that are not connected to each other. In such a case, the null eigenvalue has multiplicity K, i.e., the number of null eigenvalues is equal to the number of connected components K.

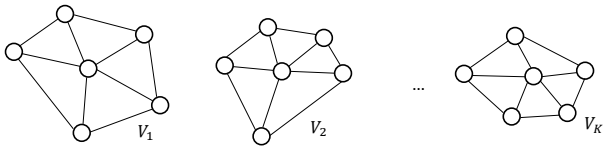

Figure 9: Graph with K sub-graphs.

Looking at the eigenvalues of the graph Laplacian provides information, not just about whether a graph is connected but also how well it is connected. The smallest non-zero

eigenvalue, called the spectral gap, informs us about the density of the graph. If a graph is densely connected (all pairs of vertices have an edge), then the spectral gap is equal to the number of vertices. Conversely, the smaller the spectral gap, the closer the graph is to being disconnected. The first positive eigenvalue thence gives a sort of continuous measure of how well the graph is connected.

The second smallest non-zero eigenvalue, called the Fiedler value (or algebraic connectivity of the graph), approximates the minimum graph cut needed to separate the graph into two connected components. Let's assume $K = 2$ and the two groups of vertices $V_1$and V_2 are linked by one additional edge, as illustrated in Figure 10. In such a case, solely looking at each value in the Fiedler vector gives an indication about which side of the cut (V_1 or V_2) the vertices v_1 and v_2 belong to.

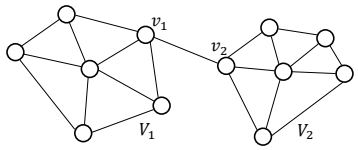

Figure 10: Two weakly connected sub-graphs.

Finally, if a graph is made of K sub-graphs, we can prove that elements of the K eigenvectors with null eigenvalues are roughly constant over each cluster. In other words, the eigenvectors' coordinates of points belonging to the same cluster are identical. Intuitively, since the eigenvectors of zero tell us how to partition the graph, the first K columns of U (the eigenvectors' coordinates) consist of the cluster indicator vectors. Let's assume two well-separated sub-graphs. The eigenvectors' coordinates $(U_{i,1}, U_{i,2})_{i=1,...,n}$ with a null eigenvalue are constant for all vertices v_i belonging to the same sub-graph V_k. Formally, for $K = 2$ we have $(U_{i,1}, U_{i,2})_{i=1,...,n} = (U_{k,1}, U_{k,2})\ \forall v_i \in V_k, k = 1, 2$. The proof is detailed in Appendix 6. If the K clusters are not identified, we can then run the k-means algorithm with rows of the first K eigenvectors as inputs representative of vertices. The full procedure is detailed in Algorithm 5.

Algorithm 5 Spectral clustering.

Initialization:

Represent the dataset X as a graph $G = (V, E, W)$

Main procedure:

1) Calculation of the $n \times n$ Laplacian matrix $L = D - A$

2) Extract the eigenvectors matrix U and diagonal matrix of eigenvalues Σ from $L = U\Sigma U^\top$

3) Fix k and build the $n \times k$ matrix $U^{(k)}$ of eigenvectors with the k eigenvalues closest to zero

4) Run the k-means algorithm 2 with the dataset of $U^{(k)}_{i,.}$ for $i = 1, ..., n$.

5) The i^{th} data point is associated to the cluster of $U^{(k)}_{i,.}$

As explained by Von Luxburg (2007), spectral clustering outperforms other popular clustering algorithms due to its ability to handle non-convex clusters. To illustrate this, we apply the spectral analysis to the circular dataset used in the introduction of this section (1200 points, 800 in the outer ring, and 400 in the inner circle). The graph is built with the mutual k-nearest neighbours for $k = 20$. The similarity parameter is $\alpha = 1$. The left plot of Figure 11 confirms that the inner and outer rings are well identified by spectral clustering. The right plot displays all the pairs of eigenvectors' coordinates $(U_{i,1}, U_{i,2})_{i=1,\ldots,1200}$. Pursuant to the above property, we observe in the right plot of Figure 11 that the eigenvectors' coordinates of points belonging to the same cluster are identical.

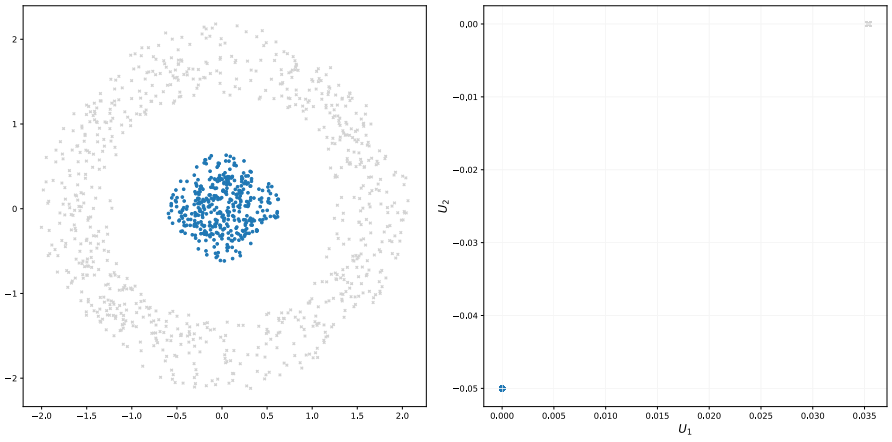

Figure 11: Left plot: partition of a non-convex dataset with spectral clustering. Right plot: pairs of eigenvectors' coordinates $(U_{i,1}, U_{i,2})_{i=1,\ldots,n}$.

In practice, implementing spectral clustering for large datasets is a challenging task, mainly because the size of the edge and weight matrices (E, W) explodes. We can eventually code (E, W) as a sparse matrix (in which most of the elements are zero) if only a few vertices are connected. Another option consists of reducing the size of the initial dataset with the k-means algorithm and then applying spectral clustering to the resulting centroids. This is a way of reducing the dimension and downplaying the high computational cost of the graph representation. The efficiency of this method is illustrated in the right plot of Figure 12. It displays the 100 centroids representative of the circular dataset of 1200 data points and their cluster. The right plot shows the pairs of eigenvectors' coordinates $(U_{i,1}, U_{i,2})$ for $i = 1$ to 100.

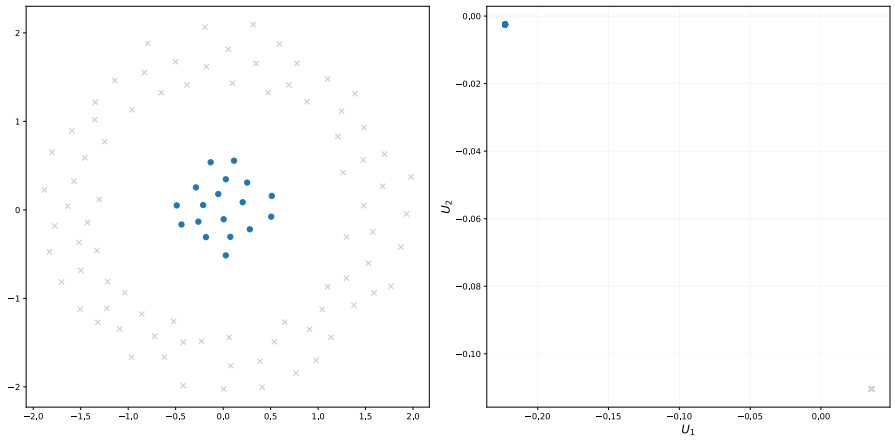

Figure 12: Left plot: spectral clustering partitioning of a non-convex dataset that has been preliminary reduced with the k-means algorithm. Right plot: pairs of eigenvectors' coordinates $(U_{i,1}, U_{i,2})_{i=1,\dots,n}$.

To conclude this section, we apply the spectral clustering algorithm to the Wasa insurance dataset. In order to work with a homogeneous distance, we once again use the conversion of the variables "driver's age" and "vehicle age" into categorical variables from sub-section 2.1.

The disjunctive table $\boldsymbol{D}$ and the weighted Burt matrix $\boldsymbol{B^W}$ are then computed using Burt's distance. As in Section 2.1, the i^{th} contract with multiple modalities is identified by the centre of gravity: $\boldsymbol{x}_i = \boldsymbol{D}_{i,.}\boldsymbol{B^W}/l$.

There are 62 436 contracts in the dataset. In order to graphically represent the dataset, we reduce its dimension by applying the k-means algorithm with 1000 centroids. The graph is then built using the method of mutual k-nearest neighbours (with k=50) and a similarity parameter ($\alpha = 1$) on this reduced dataset made up of the 1000 cluster centres. We run the spectral clustering algorithm with $K = 18$ clusters. Table 11 reports the average owner and vehicle ages, as well as the dominant features and the observed claim frequency for each cluster. We observe that the spectral clustering applied to a preliminary reduced dataset with the k-means algorithm is able to discriminate between drivers with different risk profiles. Table 10 confirms that the method achieves a reasonable goodness of fit in terms of deviance.

Goodness of fit	
Deviance	6 052.98

Table 10: Statistic of goodness of fit obtained by partitioning the dataset into 18 clusters with the spectral algorithm.

Cluster	Gender	Zone	Class	Owner's Age	Vehicle Age	Frequency (%)
1	K	4	3	[16, 28)	[12, 100)	0.8067
2	M	4	6	[16, 28)	[6, 12)	2.0656
3	M	3	3	[16, 28)	[6, 12)	1.3741
4	M	2	5	[16, 28)	[12, 100)	1.8582
5	M	4	2	[60, 99)	[6, 12)	1.1364
6	M	4	4	[16, 28)	[12, 100)	0.8716
7	M	3	3	[16, 28)	[0, 2)	4.0582
8	K	4	6	[16, 28)	[6, 12)	1.3449
9	K	4	3	[34, 54)	[12, 100)	0.6039
10	M	1	4	[30, 34)	[6, 12)	4.5135
11	M	3	3	[34, 54)	[2, 4)	1.372
12	M	3	1	[60, 99)	[12, 100)	0.7668
13	M	1	4	[34, 54)	[4, 6)	1.5576
14	M	2	3	[54, 60)	[4, 6)	0.9382
15	M	1	2	[16, 28)	[6, 12)	9.4356
16	M	1	3	[34, 54)	[12, 100)	0.8952
17	K	1	3	[34, 54)	[2, 4)	1.3468
18	M	4	3	[34, 54)	[12, 100)	0.3739

Table 11: Spectral clustering on the Wasa dataset preliminary reduced with the k-means algorithm algorithm. Dominant features and average claim frequencies per cluster.

6 Conclusions

Beyond implementing spectral clustering on an insurance portfolio, this article explored categorical data transformation to extend the scope of popular clustering techniques to actuarial applications. The main challenge consisted in defining the Burt's distance between two observations characterised by binary coded variables.

Our empirical experiment reveals that the k-means algorithm applied to Burt's distance allows us to identify relevant clusters of policies. Numerical tests emphasise the robustness of this method on fully categorised data.

The first variant of the k-means algorithm we introduced aimed at finding clusters in large-scale datasets by means of batches. The second variant was based on fuzzy logic, according to which an observation can belong to multiple clusters. Our numerical experiment reveals that the fuzzy k-means outperforms its non-fuzzy version on our dataset.

The last part of this article was devoted to spectral clustering, known as a powerful method to detect non-convex clusters. This approach requires a graphical representation that is particularly consuming in terms of computer resources when analysing large datasets. We circumvented this drawback by performing a preliminary reduction of the data using the k-means algorithm with a large number of centroids. Numerical tests carried out on a fully categorised database reveal that this approach is competitive in terms of deviance compared to methods based on standard k-means.

Acknowledgements

This work was supported by the Excellence of Science (EoS) under Grant ID 40007517.

References

[1] Mikhail Belkin and Partha Niyogi. "Laplacian eigenmaps and spectral techniques for embedding and clustering". In: *Advances in neural information processing systems* 14 (2001).

[2] Donatien Hainaut. "A self-organizing predictive map for non-life insurance". In: *European Actuarial Journal* 9.1 (2019), pp. 173–207.

[3] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[4] William H Hsu et al. "Self-organizing systems for knowledge discovery in large databases". In: *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*. Vol. 4. IEEE. 1999, pp. 2480–2485.

[5] Zhexue Huang. "Extensions to the k-means algorithm for clustering large data sets with categorical values". In: *Data mining and knowledge discovery* 2.3 (1998), pp. 283–304.

[6] Anil K Jain. "Data clustering: 50 years beyond K-means". In: *Pattern recognition letters* 31.8 (2010), pp. 651–666.

[7] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[8] T Kohonen. *Self-Organizing Maps Springer Verlag Berlin*. 1997.

[9] J MacQueen. "Classification and analysis of multivariate observations". In: *5th Berkeley Symp. Math. Statist. Probability*. 1967, pp. 281–297.

[10] Felix Mbuga and Cristina Tortora. "Spectral Clustering of Mixed-Type Data". In: *Stats* 5.1 (2021), pp. 1–11.

[11] Marina Meilă and Jianbo Shi. "A random walks view of spectral segmentation". In: *International Workshop on Artificial Intelligence and Statistics*. PMLR. 2001, pp. 203–208.

[12] Andrew Ng, Michael Jordan and Yair Weiss. "On spectral clustering: Analysis and an algorithm". In: *Advances in neural information processing systems* 14 (2001).

[13] Esbjörn Ohlsson and Björn Johansson. *Non-life insurance pricing with generalized linear models*. Vol. 2. Springer, 2010.

[14] Alberto Paccanaro, James A Casbon and Mansoor AS Saqi. "Spectral clustering of protein sequences". In: *Nucleic acids research* 34.5 (2006), pp. 1571–1580.

[15] Jianbo Shi and Jitendra Malik. "Normalized cuts and image segmentation". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.

[16] Sergei Vassilvitskii and David Arthur. "k-means++: The advantages of careful seeding". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 2006, pp. 1027–1035.

[17] Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[18] Min Wei, Tommy WS Chow and Rosa HM Chan. "Clustering heterogeneous data with k-means by mutual information-based unsupervised feature transformation". In: *Entropy* 17.3 (2015), pp. 1535–1548.

[19] Yair Weiss. "Segmentation using eigenvectors: a unifying view". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. IEEE. 1999, pp. 975–982.

[20] G Williams and Z Huang. "Mining the knowledge mine: the hot spot methodology for mining large real world data bases". In: (1997).

[21] Tengke Xiong et al. "DHCC: Divisive hierarchical clustering of categorical data". In: *Data Mining and Knowledge Discovery* 24.1 (2012), pp. 103–135.

Appendix

Proof. Meilă and Shi 2001 showed that for K weakly coupled clusters, the leading K eigenvectors will be roughly piecewise constant.

First, assume a single connected graph with K=1. Let u be an eigenvector with eigenvalue 0. Since $w_{ij} \geq 0$,

$$u'Lu = \sum_{i,j=1}^{n} w_{ij}(u_i - u_j)^2$$

is equal to 0 only if all $w_{ij}(u_i - u_j)^2 = 0$s.

This condition is met when two vertices ν_i, ν_j are disconnected (i.e., $w_{ij} = 0$). However, when two vertices ν_i, ν_j are connected (i.e., $w_{ij} > 0$) , $w_{ij}(u_i - u_j)^2 = 0$s provided that the components i and j in λ_0 are equal, which amounts to imposing $u_i = u_j$. This argument points to the conclusion that the eigenvector u needs to be constant for all vertices that can be connected by an edge in the graph. Moreover, as all vertices of a connected component (sub-graph) can be connected by an edge, u needs to be constant over the whole sub-graph. In a graph consisting of a single connected component, we thus only have the constant one eigenvector $\mathbb{1}$ with eigenvalue 0.

Consider now the case of K connected components. For merely visual purposes, we assume the vertices are ordered according to the sub-graph they belong to. Such reordering

does not affect the structure of the algorithm. Each sub-graph then corresponds to a block L_i and the Laplacian matrix L takes the form of a block diagonal matrix:

$$L = \begin{pmatrix} L_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & L_K \end{pmatrix}.$$

Each block L_i is a graph Laplacian of its own, namely the Laplacian corresponding to the sub-graph of the i^{th} connected component. We deduce from above that every L_i has eigenvalue 0 with multiplicity 1, and the corresponding eigenvector is the constant one vector on the i^{th} connected component.

Since the eigenspace of 0 gives a way to partition the graph, the blocks of positive values roughly correspond to the clusters. Given that such a matrix is constituted by a weighted sum of the outer products of eigenvectors, these eigenvectors should exhibit the property of being roughly piecewise constant (Paccanaro, Casbon and Saqi 2006). For problems with well-separated clusters, components corresponding to the elements in the same cluster should therefore have approximately the same value. By 'well-separated clusters', we mean to say the vertices in each cluster/sub-graph should be connected with high affinity (high similarity), while different clusters/sub-graphs are either not connected or are connected only by a few edges with low affinity. □