DETRA NOTE

DETRA NOTE 2022-1

INSURANCE ANALYTICS WITH K-MEANS AND EXTENSIONS

By Donatien Hainaut & Thomas Hames

Detra²lytics

DISCLAIMER

The content of the Detra Notes for a pedagogical use only. Each business case is so specific that a careful analysis of the situation is needed before implementing a possible solution. Therefore, Detralytics does not accept any liability for any commercial use of the present document. Of course, the entire team remain available if the techniques presented in this Detra Note required your attention.

Detralytics
Avenue du Boulevard 21 Box 5
1210 Brussels
www.detralytics.com
info@detralytics.eu

ABSTRACT

The k-means algorithm and its variants are popular techniques of clustering. Their purpose is to uncover group structures in a dataset. In actuarial applications, these methods detect clusters of policies with similar features and allow to draw a map of dominant risks. This working note starts with a review of the k-means algorithm and develops next two extensions to manage categorical features. We develop a mini-batch version that keeps computation time under control when analysing a high-dimensional dataset. We next introduce the fuzzy k-means in which policies can belong to multiple clusters. Finally, we conclude by a detailed introduction to spectral clustering.

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

Contents

1 Introduction

Cluster analysis is part of popular techniques within statistical data analysis and machine learning, helping to uncover group structures in data. Objects are grouped in such a way that the created groups ('clusters') are as much as possible heterogeneous between each-others, while being homogeneous regarding observations classified within them. In actuarial applications, clustering methods can detect dominant sub-populations of policies and the analysis of their claims allows a posteriori to draw a map of insured risks.

The starting point of our work is the k-means algorithm (see e.g. Mac Queen 1967, Kaufman and Rousseeuw 2009 or Hastie et al. 2009) that is one of the most popular unsupervised learning algorithms. This is a partitional method that segregates observations into an upfront specified number of clusters optimizing a measure of similarity. There exist multiple extensions of the k-means algorithm and we refer to Jain (2010) for a complete survey.

Despite its popularity in other fields such as image processing, clustering techniques are still under-exploited in actuarial science and the literature is scarce. Nevertheless, we can quote Williams and Huang (1997) who use k-means clustering to identify high claiming policyholders in a portfolio of motor vehicle insurances. Hainaut (2019) compares the k-means and self-organizing maps (SOM) to discriminate policies of motorcycle insurances. Hsu, Auvil et al. (1999) presents SOM in a framework which performs change of representation in knowledge discovery problems using insurance policy data.

The main reason explaining the under exploration of clustering techniques in actuarial science is that partitional methods exclusively manage quantitative variables in their classical version. In this case, algorithms rely on the Euclidian distance between data points to measure their similarity. Some extensions of these algorithms, aiming to take qualitative variables into account, already exist. For instance, Huang (1998) extended the k-means algorithm to data mixing quantitative and categorical variables by considering distance as a mixture of the Euclidean distance between quantitative features and a measure of dissimilarity between categorical features of two observations (Hamming's distance). Hainaut (2019) proposes as alternative a distance based on the analysis of joint frequencies (Burt's distance) of categorical variables.

The objectives of this working note are multiple. We first aim to adapt the k-means algorithm to actuarial applications. For this purpose, we study two extensions managing mixed numerical and categorical variables based on hybrid Euclidian and Hamming's or Burt's distances. We next propose a mini-batch version to perform clustering on large datasets with a limited loss of accuracy. We test the efficiency of the fuzzy k-means, a method in which policies can belong to multiple clusters. We end this work by a review of spectral clustering. As the k-means algorithm relies on the Euclidean distance, it fails to identify non-convex clusters in the space of variables. Spectral clustering exploits a deeper

data geometry based on a graphical representation of datasets. For instance, the interest reader may refer for details to Shi and Malik (2000), Ng et al. (2002) and Belkin & Niyogi (2002). We apply it to a full categorical dataset with the Burt's distance and develop a solution to manage graphs of large datasets.

2 The k-means algorithm

Let us consider a set of n numeric objects $X = \{\boldsymbol{x}_1, ...\boldsymbol{x}_n\}$ where $\boldsymbol{x}_i \in \mathbb{R}^p$ and an integer number $k \leq n$, the *k-means* algorithm searches for a partition of X into k clusters that minimises the within groups sum of squared errors (WGSS) or intraclass inertia. The k-means algorithm is based on the concept of centroids that may be interpreted as the center of gravity of a cluster of objects. 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:

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

Here,

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

is the Euclidian distance (other distances are considered in the following sections). 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

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

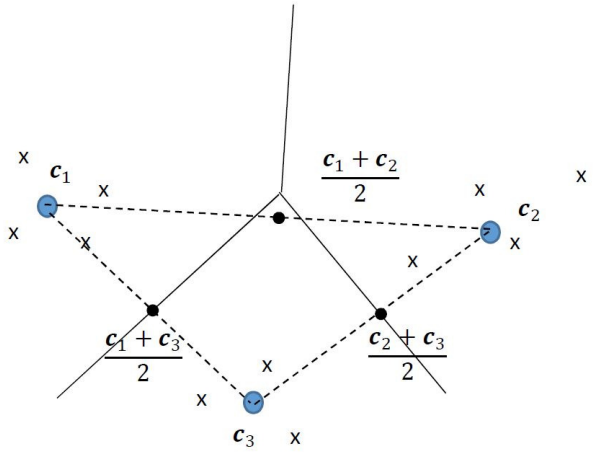

Figure 1: Illustration of the partition of a dataset with the k-means algorithm.

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:

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

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 1 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 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

Finding the partition that minimizes the intraclass inertia 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 which is detailed in Algorithm 1. 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 (1). This partition is a set of convex polyhedrons delimited by median hyperplans of centroids as illustrated in Figure 1. 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. Nevertheless, we do not have any warranty that the partition found by this way is a global solution.

The k-means 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[1]. 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 2. It improves the running time of the algorithm, and the quality of the final solution.

Algorithm 2 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

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

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

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	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 normalize the variables (we center them and divide by their standard deviation). Table 2 reports the coordinates of centroids computed with the k-means algorithm applied to variables "Owner's age" and "Vehicle age". The last column shows the claim frequency per cluster. A quick analysis reveals that the riskiest category are young drivers of less than five years old vehicles. If we use this claim frequency as predictor, noted $\widehat{\lambda}_i$, we can estimate the goodness of fit of this partition with the Poisson deviance. The deviance is the difference between log-likelihoods of the saturated model and of the the partitioned model. If N_i and ν_i are respectively the number of claims and the duration (exposure) of the i^{th} contract, this deviance is defined as:

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

The Deviance, AIC and BIC (degrees of freedom set to 20) are reported in Table 3 whereas Figure 2 displays the different clusters.

Centroids	Owner's Age	Vehicle Age	Frequency (%)
1	24	14	1.9326
2	26	4	4.3189
3	35	15	0.8366
4	41	27	0.3267
5	42	4	0.9894
6	47	46	0.1747
7	47	15	0.423
8	52	4	0.8792
9	60	17	0.2874
10	63	5	0.9963

Table 2: Coordinates of centroids and average claim frequencies for 10 clusters obtained with the k-means

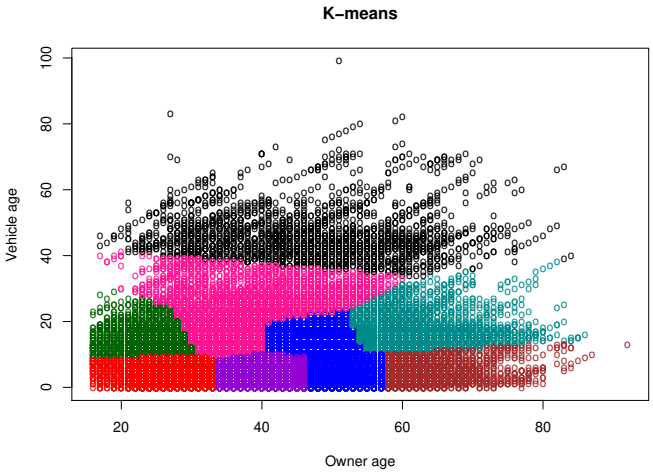

Figure 2: Illustration of the partition of a dataset with the k-means algorithm.

Goodness of fit	
Deviance	6098.82
AIC	7487.38
BIC	7668.24

Table 3: Statistics of goodness of fit obtained by partitioning the datasets in 10 clusters with the k-means.

To conclude this section, we discuss the criterions for choosing the optimal number of clusters. This choice is usually based on the marginal gain of intra-class inertia or the

marginal reduction of deviance. Above a certain number of clusters, the marginal gain of inertia/reduction of deviance is limited. This is illustrated in Figure 3 that presents inertia and deviance for various level of segmentation of the Wasa portfolio, according to variables "Owner's age" and "Vehicle age".

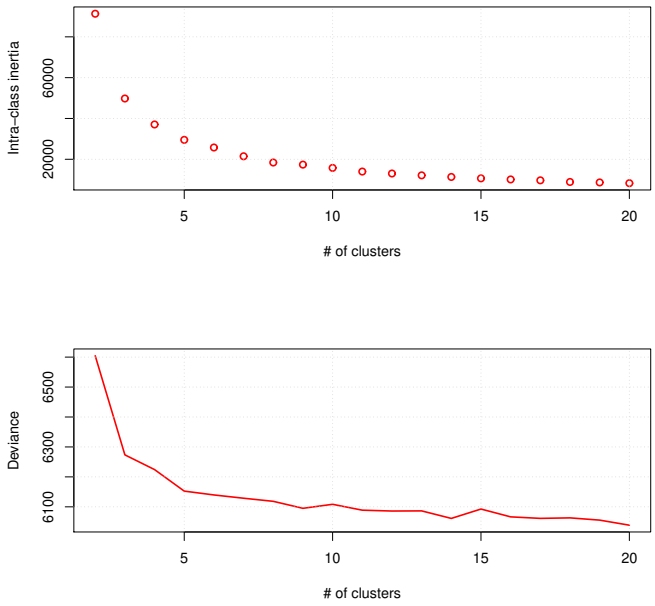

Figure 3: Upper plot: evolution of the total intra-class inertia. Lower plot: evolution of the Deviance.

3 K-means and categorical distances

As stated in the introduction, most of features of insurance policies are encoded as categorical variables. The Euclidian distance is not anymore adapted in this case. Before defining two alternative metrics, 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 p numerical variables stored in a vector $\boldsymbol{x}_{j=1,...,n} \in \mathbb{R}^p$ and l categorical variables which have m_k binary modalities for $k = 1, ..., l$. 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: $m = \sum_{k=1}^{l} m_k$. 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 exclusively 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 $l = 2$, $m_1 = 2$ 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 4.

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

Table 4: 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. Instead of running the k-means with the Euclidian distance, we will use two other metrics. The first one is the Hamming's distance that is a measure of dissimilarity between features. The second one is a measure based on the weighted Burt matrix. Both measures are based on the disjunctive table of the dataset.

3.1 Hamming's distance

The Hamming's distance between 2 policies is computed as follows:

$$d(i,j) \quad = \quad \sum_{k=1}^{m} \mathbf{1}_{\{d_{i,k} \neq d_{j,k}\}} \,,$$

where $\mathbf{1}_{\{d_{i,k} \neq d_{j,k}\}}$ is an indicator variable equal to one if $d_{i,k} \neq d_{j,k}$ and zero otherwise. If a contract has numerical and categorical features, the distance between the i^{th} and j^{th} policies is

$$d(i,j) \quad = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \quad + \beta \left(\sum_{k=1}^{m} \mathbf{1}_{\{d_{i,k} \neq d_{j,k}\}} \right), \tag{2}$$

where $\beta \in \mathbb{R}^+$ is a weight that tunes the relative importance of categorical variables with respect to numerical ones. Such an approach was proposed in Huang (1998). With this choice of metric, the second step of Algorithm 1 must be adapted. The new centroids $\boldsymbol{c}_u(e+1) = \{\boldsymbol{c}_u^p(e+1), \boldsymbol{c}_u^m(e+1)\}$ of $S_u(e)$ are defined by

- a p-vector, $\boldsymbol{c}_u^p(e+1)$, that is the center of gravity of numerical variables in a cluster

$$\frac{1}{|S_u(e)|} \sum_{\boldsymbol{x}_i \in S_u(e)} \boldsymbol{x}_i \,,$$

Centroids	Owner's age	Vehicle age	Gender	Zone	Class	Frequency
1	28	7	M	4	6	2.9208
2	29	14	M	3	2	1.9577
3	29	11	M	2	5	2.8089
4	38	10	K	3	3	0.8507
5	45	41	M	4	1	0.2329
6	46	14	M	4	4	0.4485
7	49	7	M	3	3	0.9074
8	49	15	M	4	3	0.3017
9	52	13	M	4	5	0.5189
10	54	8	M	2	1	1.4045

Table 5: Coordinates of centroids and average claim frequencies for 10 clusters with the k-means algorithm and Hamming's distance.

- a m-vector, $\boldsymbol{c}_u^m(e+1)$, of binary modalities corresponding to dominant features observed in the cluster $S_u(e)$.

Table 5 shows the result of this procedure applied to Wasa's portfolio. In addition to numerical variables "Owner's age" and "Vehicle age", we consider the categorical features: gender, zone and class. The algorithm is run with a weight $\beta = 1$. At a first sight, considering categorical variables allows us to obtain a better picture of dominant profiles in each cluster. Figure 4 shows the 10 clusters in the space Owner's and vehicle ages. We see that considering categorical variables leads to an overlap of clusters in this space. Nevertheless, the deviance, in Table 6 is slightly less good than the one with only quantitative variables. One can eventually consider to adjust β in order to minimize the deviance but we fail to find a β reducing it significantly for the Wasa dataset. Notice that we have also tested this algorithm on a dataset containing exclusively categorical variables. This dataset is built by categorizing variables "Owner's age" and "Vehicle age". Clusters obtained by this way have nevertheless a low discriminating power and the deviance is much higher than in the mixed case. In fact, the Hamming's distance is a simple measure of discordance that does not make any differences between observations that are "far" from those that are "close" to each others. This motivates us to consider another distance that is detailed in the next subsection.

Goodness of fit	
Deviance	6184.65
AIC	7633.22
BIC	8085.31

Table 6: Statistics of goodness of fit obtained by partitioning the datasets in 10 clusters with the k-means algorithm and Hamming's distance.

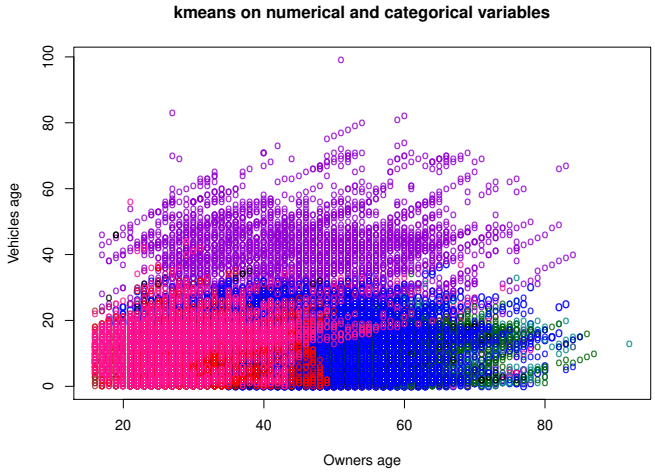

Figure 4: Illustration of the partition of a dataset with the k-means algorithm and Hamming's distance.

3.2 Burt's distance

The Hamming's distance is a simple measure of discordance between observations. Then it fails to discriminate observations that are "far" from those that are "close" to each others. The Burt's distance remedies to this issue and is based on the study of joint frequencies of modalities. 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 $\boldsymbol{B} = (n_{i,j})_{i,j=1,...,m}$ is a contingency table, called the Burt matrix containing this information. The Burt matrix is directly related to the disjunctive table as follows:

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

This symmetric matrix is composed of $l \times l$ blocks $\boldsymbol{B}_{k,j}$ for $k, j = 1, ..., l$. A block $\boldsymbol{B}_{k,j}$ is the contingency table that crosses the variables k and j. Table 7 shows the Burt matrix for the matrix $\boldsymbol{D}$ presented in Table 4. By construction, the sum of elements of a block $\boldsymbol{B}_{k,j}$ is equal to the total number of policies, n. The sum of $n_{i,j}$ of the same row i is equal to

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

The Burt matrix being symmetric, we directly infer that

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

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

In our example, we have that $n_{1,1} + n_{2,2} = n$ 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 $n_{5,5}$ 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 7: Burt matrix for the disjunctive Table 4.

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

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

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

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

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

$$n^W_{i,j} \;:=\; \frac{n_{i,j}}{\sqrt{n_{i,.}\, n_{.,j}}} \quad i,j = 1,...,m\,. \tag{3}$$

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

$$n^W_{i,j} \;:=\; \frac{n_{i,j}}{l\sqrt{n_{i,i}\, n_{j,j}}} \quad i,j = 1,...,m\,. \tag{4}$$

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 $\boldsymbol{B^W}$:

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

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

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

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

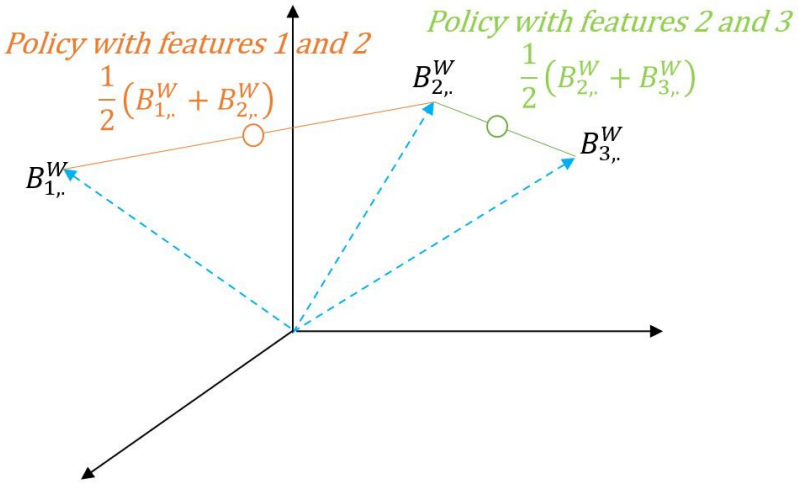

Figure 5: Illustration of the partition of a dataset with the k-means algorithm.

The k^{th} modality corresponds then to the k^{th} line of $\boldsymbol{B^W}$, a vector in $\mathbb{R}^m$. The i^{th} contract with multiple modalities can then be identified by the center of gravity $\boldsymbol{D}_{i,.}\boldsymbol{B^W}/l$, of points with coordinates stored in the corresponding lines of the weighted Burt matrix. This point is illustrated in Figure 5 in the case of three modalities. If each policy is defined by a subset of l =2 modalities, we represent in $\mathbb{R}^3$ as the mid point between corresponding lines of $\boldsymbol{B^W}$. The mixed Euclidian and Burt's distance between the i^{th} and j^{th} policies is then

$$d(i,j) \quad = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \quad + \beta \left\|\boldsymbol{D}_{i,.}\boldsymbol{B^W}/l - \boldsymbol{D}_{j,.}\boldsymbol{B^W}/l\right\|_2 , \tag{5}$$

where $\beta \in \mathbb{R}^+$ is a weight. The second step of Algorithm 1 must be adapted. The new centroids $\boldsymbol{c}_u(e+1) = \{\boldsymbol{c}_u^p(e+1), \boldsymbol{c}_u^m(e+1)\}$ of $S_u(e)$ are defined by

- a p-vector, $\boldsymbol{c}_u^p(e+1)$, that is the center of gravity of numerical variables in a cluster

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

- a m-vector, $\boldsymbol{c}_u^m(e+1)$, that is the categorical counterpart:

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

At the end of the procedure, we identify the dominant modalities in each clusters as the most frequent ones. We run this algorithm with a parameter $\beta = 10$. This value is chosen because it leads to the lowest deviance. Table 7 provides the results with 10 centroids. The youngest male drivers of a recent motorbike are still identified as the riskiest category of insureds. The male drivers that are 45 years old and driving ancestor motorcycles have the lowest claim frequency. The deviance, reported in Table 9 is slightly better than the one of a segmentation based only on owner's and vehicle ages. Figure 4 shows the 10 clusters in the space Owner's and vehicle ages. As for the Hamming's distance, we see that considering categorical variables leads to an overlap of clusters in this space.

Centroids	Owner's age	Vehicle age	Gender	Zone	Class	Frequency (%)
1	26	5	M	4	3	4.2638
2	28	15	M	2	5	1.9384
3	31	16	M	4	5	0.7463
4	45	43	M	4	1	0.2541
5	46	4	M	4	3	0.8951
6	46	10	K	4	3	0.682
7	49	16	M	2	3	0.5918
8	50	16	M	3	3	0.3894
9	52	17	M	4	3	0.2917
10	62	6	M	4	3	0.757

Table 8: k-means algorithm and Burt's distance. Average owner's and vehicle ages, dominant features and average claim frequencies per cluster.

Goodness of fit	
Deviance	6082.60
AIC	7751.17
BIC	9197.87

Table 9: Statistics of goodness of fit obtained by partitioning the datasets in 10 clusters with the k-means algorithm and Burt's distance.

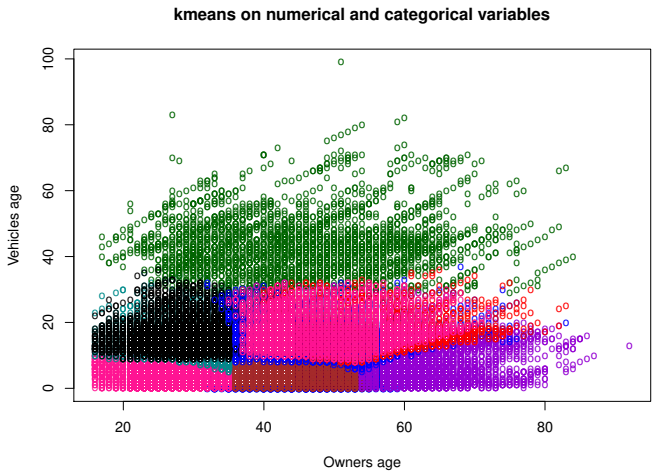

Figure 6: Illustration of the partition of a dataset with the k-means algorithm with Burt's distance.

To conclude this section we apply the k-means to the Wasa insurance dataset fully converted into categorical variables. The distance in this case is homogeneous and exclusively evaluated with the weighted Burt's table. For this purpose, we convert the variables "driver's age" and "vehicle age" into categorical variables with 6 modalities. Categories are designed with the k-means algorithm. Next we compute the matrix of $\boldsymbol{x}_i = \boldsymbol{D}_{i,\cdot}\boldsymbol{B}^{\boldsymbol{W}}/l$ for $i = 1$ to n and apply the standard k-means algorithm. Table 10 reports the mean driver's and vehicle ages, the most frequent features and the claim frequency in each cluster. This allows to quickly detect the most and the less risky driver's profiles. The deviance (Table 9) is comparable to the one obtained with other approaches. In comparison, the deviance of a GLM model fitted to the same dataset is around 5790 whereas the deviance of the null model is equal to 6648. Notice that the AIC and BIC are computed with a number of degrees of freedom equal to 20×28 (# of clusters × # of modalities). At a first sight, the AIC in Table 9 seems less good that the one of previous models. This would be true if datasets were the same. The increase of AIC is here mechanically linked to the conversion to categorical variables.

Owner's age	Vehicle age	Gender	Zone	Class	Frequency (%)
24	7	M	3	3	5.56
32	10	M	1	3	2.71
36	11	K	4	3	1.38
39	9	K	4	3	0.98
39	16	K	3	3	0.35
41	11	M	4	6	1.58
41	16	K	4	3	0.47
42	2	K	4	3	1.23
42	12	M	4	4	0.61
43	5	M	2	3	0.98
43	13	K	4	3	0.92
43	17	M	3	3	0.43
44	12	M	4	3	0.49
45	17	M	3	5	0.66
45	17	M	4	3	0.3
47	47	M	4	1	0.2
48	22	M	4	1	0.33
51	6	M	4	5	0.72
52	4	M	4	3	0.92
59	10	M	2	3	1.03

Table 10: Average owner's and vehicle ages, dominant features and average claim frequencies per cluster. k-means applied to a full categorical dataset.

Goodness of fit	
Deviance	6083.61
AIC	8552.18
BIC	13615.65

Table 11: Statistics of goodness of fit obtained by partitioning the full categorical dataset in 20 clusters with the k-means.

4 Mini-batch k-means

For large datasets, the computation time of k-means increases because of its constraint of needing the whole dataset in main memory. For this reason, several methods have been proposed to reduce the temporal and spatial cost of the algorithm. A different approach is the Mini batch k-means algorithm.

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

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

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}:

$$\boldsymbol{c}_u^{p\,new} = \frac{1}{|S_u^{new}|} \sum_{i \in S_u^{new}} \boldsymbol{x}_i\,,$$

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

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

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

End loop on centroids, u.

End loop on epochs e

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

Mini Batch k-means algorithm's main idea is to use small random batches of data with a fixed size, so they can be stored in memory. Each iteration a new random sample from the dataset is obtained and used to update the clusters, taking care of deprecating previous coordinates according to a learning speed. This operation is repeated until convergence. The algorithm 3 presents the details of this approach for mixed numerical and categorical variables combined with the Burt's distance.

The empirical results in the literature suggest that it can obtain a substantial saving of

Owner's age	Vehicle age	Gender	Zone	Class	Frequency
26	6	M	4	3	4.2518
28	10	K	4	3	1.2647
29	17	M	4	5	1.3413
45	6	M	4	4	0.9823
45	17	M	4	3	0.3726
45	43	M	4	1	0.2492
46	4	M	4	3	0.8167
46	14	K	4	3	0.6454
59	5	M	4	3	0.8883
59	17	M	4	3	0.2662

Table 12: Coordinates of centroids and average claim frequencies for 10 clusters, mini-batch k-means.

computational time at the expense of some loss of cluster quality, but not extensive study of the algorithm has been done to measure how the characteristics of the datasets, such as the number of clusters or its size, affect the partition quality. As the number clusters and the number of data increases, the relative saving in computational time also increases.

We run this algorithm with the Burt's distance (5), a parameter $\beta = 10$ and batches of 10 000 policies. We recall that β is chosen because it leads to the lowest deviance. Table 12 provides the results with 10 centroids. We retrieve most of categories found in Sub-section 3.2 except that we have now two categories with female drivers. The deviance, reported in Table 13 is slightly better than the one of a segmentation based only on owner's and vehicle ages.

Notice that the AIC and BIC are computed with a number of degrees of freedom computed as # of clusters $\times$ # of modalities. This explains why AIC and BIC mechanically increase due to the categorization of numerical variables. Figure 7 shows the 10 clusters in the space Owner's and vehicle ages. In our case study, the gain of computation time with respect to k-means is nevertheless limited (both algorithms run in a few seconds). To observe significant gains of computation time, the mini-batch k-means should be tested on a larger dataset than the one of Wasa.

Goodness of fit	
Deviance	6077.27
AIC	7785.84
BIC	9413.38

Table 13: Statistics of goodness of fit obtained by partitioning the datasets in 10 clusters with the mini-batch k-means.

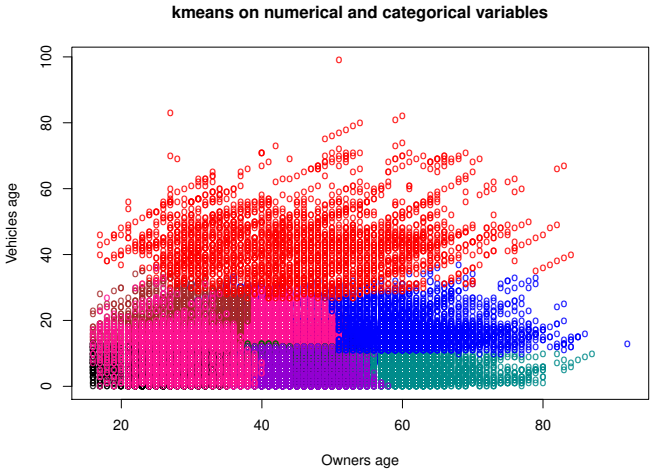

Figure 7: Illustration of the partition of a dataset with the mini-batch k-means algorithm with Burt's distance.

5 Fuzzy k-means

In non-fuzzy clustering (also known as hard clustering), data is divided into distinct clusters, where each data point can only belong to exactly one cluster. In fuzzy clustering, data points can potentially belong to multiple clusters. The algorithm minimizes intra-cluster variance as well, but has the same problems as k-means; the minimum is a local minimum, and the results depend on the initial choice of weights.

The fuzzy k-means algorithm attempts to partition a finite collection of n elements into a collection of k fuzzy clusters, S_u for $u = 1, ..., k$, with respect to some given criterion. 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 $w_{i,j}$ tells the degree to which the i^{th} policy belongs to cluster S_j. The fuzzy k-means aims to minimize an objective function:

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

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. The fuzzifier, m, determines the level of cluster fuzziness. A large m results in smaller membership values $w_{i,j}$,

and hence, fuzzier clusters. In the limit $m = 1$, the memberships, $w_{i,j}$, converge to 0 or 1, which implies a crisp partitioning. The fuzzy k-means is detailed in algorithm 4.

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 $\boldsymbol{c}_u(e+1) = (\boldsymbol{c}_u^p(e+1), \boldsymbol{c}_u^m(e+1))$ of $S_u(e)$:

$$\boldsymbol{c}_u^p(e+1) = \frac{\sum_{i=1}^{n} w_{i,u}(e)^m \boldsymbol{x}_i}{\sum_{i=1}^{n} w_{i,u}(e)^m}$$

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

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

Owner's age	Vehicle age	Gender	Zone	Class	Frequency (%)
23	10	M	4	6	2.86
26	6	M	4	3	5.25
31	10	M	2	4	2.22
35	10	M	2	6	2.98
38	10	K	4	6	1.3
40	10	K	2	3	0.93
40	15	K	4	5	0.52
41	8	K	4	4	1.3
41	10	K	4	3	0.5
42	12	M	2	3	0.89
44	15	M	4	5	0.52
45	13	M	4	4	0.46
45	16	M	4	3	0.32
45	16	M	3	3	0.53
46	8	M	1	3	1.99
47	11	M	4	3	0.44
48	33	M	4	1	0.42
49	10	M	4	5	0.76
49	10	M	2	3	0.67
57	9	M	2	3	1.14

Table 14: Fuzzy clustering. Average owner's and vehicle ages, dominant features and average claim frequencies per cluster.

We apply the 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 with 6 modalities as in the Sub-section 3.2. We run the algorithm with a fuzziness parameter equal to $m = 1.10$. Table 14 reports the mean driver's and vehicule ages, the most frequent features and the claim frequency in each cluster. The policies are assigned to the most likely cluster (highest $w_{i,j}$ for $j = 1, ..., k$). We find similar most and less risky driver's profile to those of Sub-section 3.2. The deviance (Table 15) is nevertheless less good than with other approaches. 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 well a non-null probability to belong to them.

Goodness of fit	
Deviance	6 135.27
AIC	8 603.84
BIC	13 667.31

Table 15: Fuzzy clustering. Statistics of goodness of fit obtained by partitioning the full categorical dataset in 20 clusters.

6 Spectral clustering

As the k-means algorithm relies on the Euclidean distance, it does not perform well on non-convex geometrical representation. To illustrate this, we apply the k-means to partition the circular dataset plotted in the left plot of Figure 8 into 2 or 4 clusters. Clusters obtained by this way are shown in the mid and right plots. The algorithm clearly fails to identify the inner and outer rings.

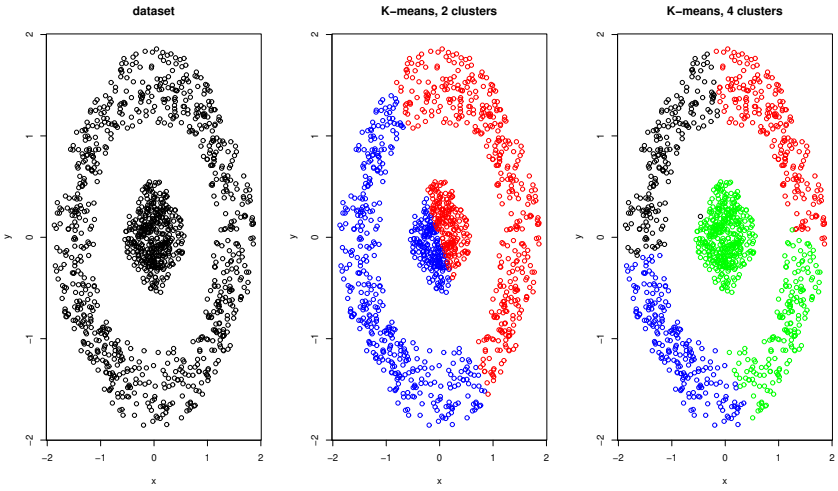

Figure 8: Illustration of the partition of a non-convex dataset with the k-means algorithm. Each cluster is identified by a colour.

One solution to cluster non-convex shape consists to exploit a deeper data geometry. It is feasible through spectral clustering (Shi and Malik, 2000, Ng et al., 2002, Belkin & Niyogi, 2002). It works by embedding the data in a different space derived from a graphical representation of the dataset. Applying k-means on this representation allows identifying non-convex clusters.

Spectral clustering uses the graph theory to represent the data points. As illustrated in Figure 9, a graph G is defined by three elements: vertices v_i representing data points, edges e_{ij} representing link between vertices v_i and v_j and weights w_{ij}. These weights are for instance the distance between two vertices linked by an edge.

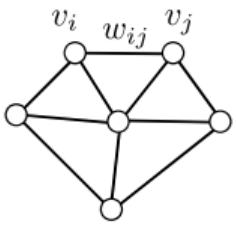

Figure 9: Vertices, edges and weights of a graph.

Mathematically, a graph G is defined as $G = (V, E, W)$ with V, E and W being respectively the set of all Vertices, Edges and Weights. If we denote $v_i \longleftrightarrow v_j$ when an edge links v_i to v_j, the graph is represented by:

$$\begin{aligned} V &= \{v_i\}_{i=1}^{n} \\ E &= \{e_{i,j} : v_i \longleftrightarrow v_j\} \\ W &= \{(w_{ij} : w_{ij} \neq 0 \text{ if } v_i \longleftrightarrow v_j\} \end{aligned}$$

Elements E and W can be represented as $n \times n$ matrices where n is the number of data points in the dataset. The elements $e_{i,j}$ of E are equal to 1 if $v_i \longleftrightarrow v_j$ and 0 otherwise. The matrix W contains distance w_{ij} if two vertices i and j are linked by an edge.

In order to work with these representations, we use a kind Fourier transformation on the graph G based on its Laplacian representation. The Laplacian representation of a graph G is defined as:

$$L = D - W$$

where D being a diagonal matrix with diagonal elements that are $D_{ii} = \sum_j w_{i,j}$. D is often referred as the degree matrix.

Why is matrix L called Laplacian? We can define a function on a graph, $f : V \to \mathbb{R}$ such that $v_i \to f(v_i)$. Let us consider a discrete periodic function which takes N values, at times $1, 2, ..., N$. The loop on periods may be represented by a ring graph as shown in Figure 10.

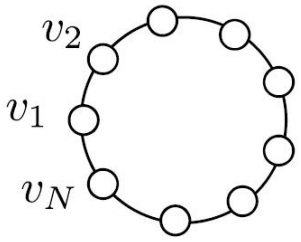

Figure 10: Ring representation of a period with N steps.

The matrix of edges and weights is in this particular case

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

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

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

We can next extract the eigenvalues and eigenvectors of the Laplacian (spectral analysis). Since L is symmetric, we can rewrite Laplacian L as $L = U\Sigma U^\top$ where U is a matrix containing all the eigenvectors and Σ a diagonal matrix containing the eigenvalues.

Analyzing eigenvalues extracted from a graph provides useful information about its structure. For instance, if all vertices of a graph are completely disconnected, all eigenvalues are null. As we add edges, some of the eigenvalues becomes non-null. The number of null eigenvalues corresponds to the number of groups of connected vertices in our graph. As illustration, let us consider the graph plotted in Figure 11 that is made of K different groups of vertices not connected between each others.

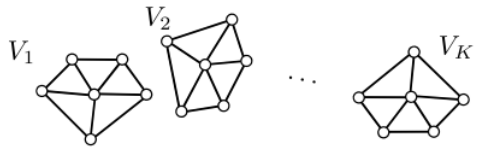

Figure 11: Graph with K sub-graphs.

In such a case, the number of eigenvalues equal to 0 is equal to the number of groups K. If all the vertices are connected between each-others, we would only observe one group and thus only one eigenvalue equals to 0. Moreover, the first nonzero eigenvalue, called the

spectral gap, informs us about of the density of the graph. If a graph is densely connected (all pairs of the nodes have an edge), then the spectral gap is equal to the number of vertices.

The second eigenvalue is called the Fiedler value, and the corresponding vector is the Fiedler vector. The Fiedler value approximates the minimum graph cut needed to separate the graph into two connected components. Let us imagine that in our example that $K = 2$ and that the 2 groups of vertices are linked by one additional edge. For such case, simply by looking at each value in the Fiedler vector, it would give us information about which side of the cut (V_1 or V_2) that vertice belongs.

Finally, if a graph is made of K sub-graphs, we can prove that elements of the K eigenvectors with null eigenvalues are constant over each cluster, as illustrated in Figure 12. If the K clusters are not identified, we can run the k-means algorithm with rows of the first K eigenvectors as input representative of vertices. The full procedure is detailed in algorithm 5.

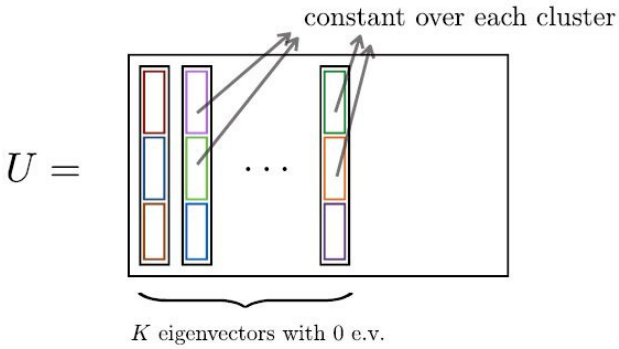

Figure 12: Matrix of eigenvectors of a the Laplacian of a graph with K sub-graphs.

At this stage, we haven't discussed yet how to represent an initial dataset as a graph. The first step consists to associate vertices $(v_j)_{j=1...n}$ to each data points. We next define a measure of similarity that is inversely proportional to the distance between two data points. This similarity is used to construct the graph. Two highly similar data points will be connected by an edge with a weight equal to their similarity measure. At the opposite, data points with a low similarity are considered as disconnected. A common measure of similiarity is based on a Gaussian kernel:

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

where $\alpha \in \mathbb{R}^+$ is a tuning parameter. There exist different ways of creating the pairwise similarities graphs representation and this choice generally has an influence on the created clusters.

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

$$L = D - W$$

2) Compute the eigenvectors matrix U and diagonal matrix of eigenvalues Σ of L, such that

$$L = U \Sigma U^\top$$

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

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

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

a. The ϵ-neighborhood graph: we connect all the points for which the pairwise distances are smaller than ϵ. In practice, it means keeping all distances/similarities smaller than ϵ and forcing all the others to 0.

b. The k-nearest neighbor graph: we connect the vertex v_i with the vertex v_j if v_j is among the k-nearest neighbors of v_i. However, the neighborhood relationship is not symmetric and we need the graph to be symmetric. There exist two ways to force the symmetry: the first one is simply to ignore the directions of the edges, meaning that we connect v_i and v_j if v_i is among the k-nearest neighbors of v_j **or** if v_j is among the k-nearest neighbors of v_i. The resulting graph is usually called the k-nearest neighbors graph. The second way is to connect vertices v_i and v_j if both v_i is among the k-nearest neighbors of v_j **and** v_j is among the k-nearest neighbors of v_i. The resulting graph is usually called the mutual k-nearest neighbors graph.

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

As explained by Von Luxburg (2007), the spectral clustering outperforms other popular clustering algorithms due to its ability to handle non-convex clusters. To illustrate this, we apply this method to the circular dataset used in the introduction of this section (1200 points, 800 in an outer ring and 400 in the central circle). We build the graph with the mutual k-nearest neighbors for $k = 20$. The similarity parameter is $\alpha = 1$. The left plot of Figure 13 confirms that inner and outer rings are well identified by spectral clustering. The right plot shows all the pairs of eigenvector coordinates $(U_{i,1}, U_{i,2})_{i=1,...,1200}$. We observe that coordinates of points belonging to the same cluster are identical.

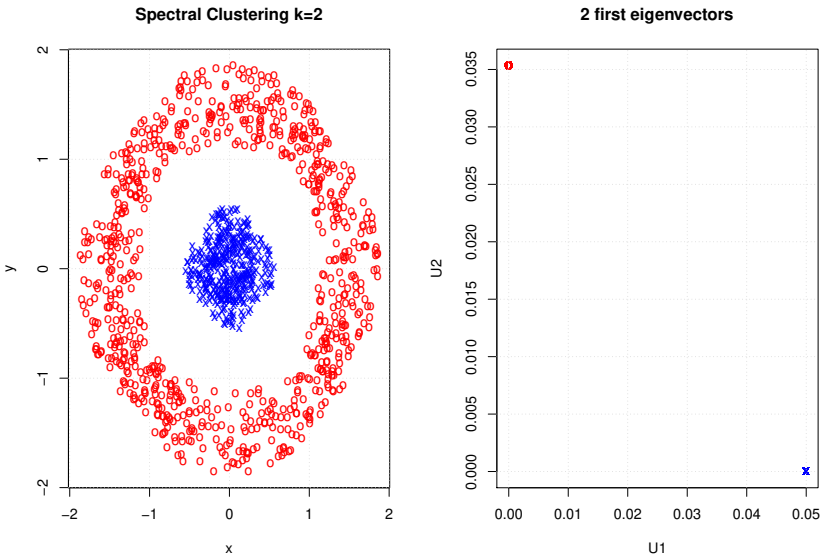

Figure 13: Left plot: partition of a non-convex dataset with spectral clustering. Right plot: pairs of eigenvector coordinates $(U_{i,1}, U_{i,2})$ for $i = 1$ to n.

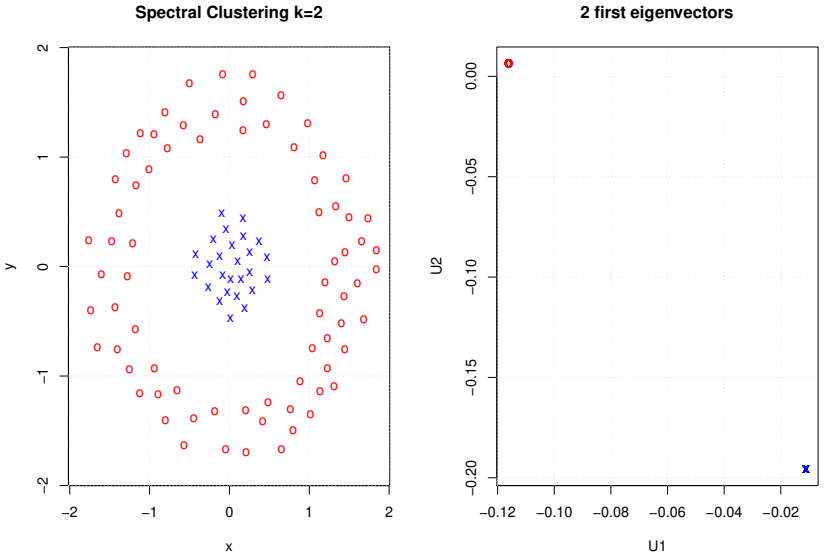

Figure 14: Left plot: partition of a non-convex dataset with spectral clustering preliminary reduced with the k-means algorithm. Right plot: pairs of eigenvector coordinates $(U_{i,1}, U_{i,2})$ for $i = 1$ to n.

In practice, implementing spectral clustering for large datasets is a challenging task mainly because the sizes of edge and weight matrix (E, W) explode. We can eventually code (E, W) as sparse matrix if few vertices are connected. An alternative consists to reduce the size of the initial dataset with the k-means algorithm and to apply spectral clustering to centroids. The efficiency of this method is illustrated in the right plot of Figure 14. It displays the 100 centroids representative of the circular dataset of 1200 data points and their cluster. The right plot shows the pairs of eigenvector coordinates $(U_{i,1}, U_{i,2})$ for $i = 1$ to 100.

To conclude this section we apply spectral clustering to the Wasa insurance dataset. In order to work with homogeneous distance, we convert the variables "driver's age" and "vehicle age" into categorical variables with 6 modalities as in the second example of sub-section 3.2. Categories are designed with the k-means algorithm.

We use the Burt's distance and compute the disjunctive table $\boldsymbol{D}$ and the weighted Burt matrix denoted by $\boldsymbol{B^W}$. As in Section 3.2, the i^{th} contract with multiple modalities is identified by the center of gravity $\boldsymbol{x}_i = \boldsymbol{D}_{i,.}\boldsymbol{B^W}/l$, of points with coordinates stored in the corresponding lines of the weighted Burt matrix. The dataset counts 62 436 contracts and we have to reduce its dimension in order to graphically represent the dataset. For this reason, we apply the k-means algorithm with 1500 centroids. The graph is next built with the method of mutual k-nearest neighbours (with k=20) and a similarity parameter ($\alpha = 1$) applied to centroids. We run the spectral clustering algorithm with $k = 20$ clusters. Table 17 reports the average owner's and vehicle ages, the dominant features and the observed claim frequency for each cluster. We see that this method is able to discriminate drivers with different risk profiles. Table 16 confirms that the method achieves a reasonable goodness of fit in term of deviance. The AIC and BIC are computed with a number of degrees of freedom equal to 20×20 (# of clusters × # of eigenvectors).

Goodness of fit	
Deviance	6089.843
AIC	8238.413
BIC	11855.17

Table 16: Statistics of goodness of fit obtained by partitioning the datasets in 20 clusters with the spectral algorithm.

Owner's age	Vehicle age	Gender	Zone	Class	Frequency
25	4	M	1	4	7.54
25	10	M	4	6	3.2
26	9	K	2	4	1.65
38	16	K	3	5	0.41
41	10	K	4	3	0.51
41	10	K	1	4	0.62
42	9	K	1	3	0.91
42	16	M	3	5	0.71
42	25	M	4	3	0.46
43	10	M	3	5	1.54
43	11	K	4	4	1.07
44	10	K	4	6	1.61
45	26	K	2	1	0.94
45	32	M	3	1	0.77
46	8	K	2	4	0.94
46	11	M	3	4	0.71
47	12	M	4	3	0.47
48	5	M	3	3	1.14
52	16	M	2	2	1.39
59	7	M	3	4	1.21

Table 17: Categories obtained with the spectral clustering algorithm.

7 Conclusions

This working note explores the potential applications of popular clustering techniques to insurance datasets. The main challenge consists to define a distance between two observations characterized by categorical and numerical variables. Two approaches are proposed. The first one mixes the Euclidian and Hamming's distances, that is a measure of similarity. Our empirical experiment reveals that combining this hybrid distance with the k-means allows identifying relevant clusters of policies. Nevertheless, the Hamming's distances has a low discriminating power mainly because it does not make difference between "far" and "close" observations. As alternative, we consider a Burt's distance based on the analysis of joint frequencies. Numerical tests emphasize the robustness of this method both on hybrid numerical-categorical and full categorical datasets.

We next presents two interesting variants of the k-means algorithm. The first one is based on batches that allows to find clusters in large datasets. The other one is based on fuzzy logic: an observation can belongs to multiple clusters. Our numerical experiment reveals that the fuzzy k-means do not outperform its non-fuzzy version on our dataset.

The last part of this article is devoted to spectral clustering that is a powerful method to detect non-convex clusters. Nevertheless, this approach requires a graphical representation that is particularly consuming in terms of computer resources when analyzing a large dataset. We circumvent this drawback by a preliminary reduction of data using a k-means procedure with a high number of centroids. Numerical tests carried on a full categorical database reveals that this approach is competitive in term of deviance compared to methods based on standard k-means.

References

[1] Arthur D., 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] Belkin M, & Partha N., 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. Pages 585–591 of: Advances in Neural Information Processing Systems 14. MIT Press.

[3] Hainaut D. 2019. Self-Organizing Maps for non-life insurance. European Actuarial Journal, 9, pp 173–207.

[4] Hastie T., Tibshirani R., Friedman J. 2009. The Elements of Statistical Learning. Data Mining, Inference, and Prediction. 2nd edition. Springer Series in Statistics.

[5] Hsu W., Auvil, L., Pottenger W.M., Tcheng D., Welge M., 1999. Self-Organizing Systems for Knowledge Discovery in Databases. Proceedings of the International Joint Conference on Neural Networks (IJCNN-1999)

[6] Huang Z. 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values Data Mining and Knowledge Discovery 2, 283–304

[7] Jain A.K. 2010. Data Clustering: 50 Years Beyond K-Means. Pattern Recognition Letters. 31 (8), 651-666.

[8] Kaufman L.,Rousseeuw P.J. 2009. Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, vol. 344.

[9] Kohonen, T. 1997. Self-organizing maps. Springer, New York.

[10] MacQueen J. 1967. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297.

[11] Ng A.Y., Jordan M., Wess Y., 2001. On spectral clustering: Analysis and an algorithm. Pages 849–856 of: Advances in Neural Information Processing Systems 14. MIT Press.

[12] Ohlsson E., Johansson B., 2010. Non-life insurance pricing with generalized linear models. Springer-Verlag Berlin Heidelberg

[13] Shi J., Malik J. 2000. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence, 22(8), 888–905.

[14] Von Luxburg U.. A tutorial on spectral clustering. Statistics and computing, 17(4):395–416, 2007.

[15] Williams G., Huang Z. 1997. Mining the Knowledge Mine: The Hot Spots Methodology for Mining Large Real World Databases. Proceedings of the 10th Australian Joint Conference on Artificial Intelligence: Advanced Topics in Artificial Intelligence, pp. 340–348.

8 About the serie and the authors

8.1 The DetraNotes

The Detra Notes are a series of educational papers dedicated to the insurance sector. Those notes are published by members of the Detralytics team and written in a clear and accessible language. The team combines academic expertise and business knowledge. Detralytics was founded to support companies in the advancement of actuarial science and the solving of the profession's future challenges. It is within the scope of this mission that we make our work available through our DetraNotes and FAQctuary's series.

8.2 Authors' biographies

Donatien Hainaut

Donatien Hainaut is Scientific Director at Detralytics and professor at UCLouvain where he is Director of the new Master program in Data Science, statistical orientation. Prior to this he held several positions as associate professor at Rennes School of Business and the ENSAE in Paris. He also has several field experiences having worked as Risk Officer, Quantitative Analyst and ALM Officer.

Donatien is a Qualified Actuary and holds a PhD in the area of Assets and Liability Management. His current research focuses on contagion mechanism in stochastic processes and applications of neural networks to insurance.

Thomas Hames

Thomas is part of the Talent Consolidation Program (TCP) at Detralytics. Prior to joining Detralytics, Thomas worked as an intern at AXA in the P&C Retail department and developped a Geo-Spatial analysis based on Machine Learning models. Thomas holds a Bachelor's degree in Business Engineering from UCLouvain and a Master's degree in Actuarial Sciences from UCLouvain. His master thesis focused on the modelling and forecasts of the mortality rates for the Belgium.

Expertise and innovation at the service of your future

www.detralytics.com - info@detralytics.eu