Volume 33 (2), pp. 85–103
http://orion.journals.ac.za

ORiON
ISSN 0259–191X (print)
ISSN 2224–0004 (online)

A semi-supervised segmentation algorithm as applied to k-means using information value

DG Breed* T Verster† SE Terblanche‡

Received: 15 September 2016; Revised: 20 April 2017; Accepted: 8 May 2017

Abstract

Segmentation (or partitioning) of data for the purpose of enhancing predictive modelling is a well-established practice in the banking industry. Unsupervised and supervised approaches are the two main streams of segmentation and examples exist where the application of these techniques improved the performance of predictive models. Both these streams focus, however, on a single aspect (*i.e.* either target separation or independent variable distribution) and combining them may deliver better results in some instances. In this paper a semi-supervised segmentation algorithm is presented, which is based on k-means clustering and which applies information value for the purpose of informing the segmentation process. Simulated data are used to identify a few key characteristics that may cause one segmentation technique to outperform another. In the empirical study the newly proposed semi-supervised segmentation algorithm outperforms both an unsupervised and a supervised segmentation technique, when compared by using the Gini coefficient as performance measure of the resulting predictive models.

Key words: Banking, clustering, multivariate statistics, data mining.

1 Introduction

The use of segmentation within a predictive modelling context is a well-established practice in the industry [2, 47, 55]. The ultimate goal of any segmentation exercise is to achieve more accurate, robust and transparent predictive models [55]. The focus of this paper is on extending the available techniques that can be used for statistical segmentation, for the purpose of improving predictive power. Two main streams of statistical segmentation are used in practice, namely unsupervised and supervised segmentation.

*Centre for Business Mathematics and Informatics, North-West University, Potchefstroom, South Africa
†Corresponding author: Centre for Business Mathematics and Informatics, North-West University, Potchefstroom, South Africa, email: Tanja.Verster@nwu.ac.za
‡Centre for Business Mathematics and Informatics, North-West University, Potchefstroom, South Africa

http://dx.doi.org/10.5784/33-2-568

Unsupervised segmentation [22] maximises the dissimilarity of the character distributions of segments based on a distance function. The technique focusses on the independent variables in a model and does not take the target variable into account. A popular example of unsupervised segmentation is clustering.

Supervised segmentation maximises the target separation or impurity between segments [24]. The technique focusses, therefore, on the target variable and not on identifying subjects with similar independent characteristics. A very popular example of supervised segmentation is a decision tree.

Both these streams make intuitive sense depending on the application and the requirements of the models developed [19] and many examples exist where the use of either technique improved model performance [21]. Both these streams focus, however, on a single aspect (*i.e.* either target separation or independent variable distribution) and combining both aspects may deliver better results in some instances.

In this paper a semi-supervised segmentation algorithm is proposed as an alternative to the segmentation algorithms currently in use. This algorithm will allow the user, when segmenting for predictive modelling, to not only consider the independent variables (as is the case with unsupervised techniques such as clustering) or the target variable (as is the case with supervised techniques such as decision trees), but to be able to optimise both during the segmentation approach. The unsupervised component of the newly proposed algorithm is based on k-means clustering and information value [34] is used as a measure of the separation, or impurity, of the target variable.

Simulated data are used to identify which characteristics may cause one segmentation technique to outperform another when segmenting for predictive modelling. Furthermore, empirical results are provided to showcase the performance of the newly proposed semi-supervised segmentation algorithm.

The outline of the paper is as follow: Section 2 starts with a literature review of segmentation techniques, focussing specifically on segmentation within the predictive modelling context. Section 3 provides the necessary definitions and notations and in Section 4 details of the newly proposed semi-supervised segmentation algorithm are provided. In Section 5, empirical results are provided for the purpose of comparing the newly proposed algorithm with a supervised and an unsupervised segmentation approach. Section 6 concludes and discusses further research ideas.

2 Literature review

A multitude of analytic methods are associated with data mining and they are usually divided into two broad categories: pattern discovery and predictive modelling [25].

Pattern discovery usually involves the discovery of interesting, unexpected, or valuable structures in large data sets using input variables. There is usually no target/label in pattern discovery and for this reason it is sometimes referred to as unsupervised classification. Pattern discovery examples include segmentation, clustering, association, and sequence analyses.

Predictive modelling is divided into two categories: continuous targets (or labels) and discrete targets (or labels). In predictive modelling of discrete targets, the goal is to assign discrete class labels to particular observations as outcomes of a prediction. This is commonly referred to as supervised classification, and in this context predictive modelling is sometimes also referred to as supervised classification. Predictive modelling examples include decision trees, regression, and neural network models.

Segmentation of data for the purpose of building predictive models is a well-established practice in the industry. Siddiqi [47] divides segmentation approaches into two broad categories, namely experience-based (heuristic) and statistically based. Hand [24] split the methods of segmentation into two groups as discussed in the introduction: unsupervised and supervised.

Popular unsupervised segmentation techniques include clustering (*e.g.* k-means, density based or hierarchical clustering); hidden Markov models [9] and feature extraction techniques such as principal component analysis. Of these techniques, the one most commonly used for segmentation is clustering. Cluster analysis traces its roots back to the early 1960s [50] and it was the subject of many studies from the early 1970s [1, 10]. K-means clustering is one of the simplest and most common clustering techniques used in data analysis. It follows a very simple iterative process that continuously cycles through the entire data set until convergence is achieved.

Density based clustering makes use of probability density estimates to define dissimilarity as well as cluster adjacency [28, 61]. In contrast to the k-means algorithm, these clustering techniques do not start off with a pre-defined number of clusters, but is agglomerative in that it starts with each observation in its own cluster. Clusters are then systematically combined to minimise the dissimilarity measure used. Computationally, these techniques are significantly more complex than k-means clustering but possess the ability to form clusters of any form and size [24]. The k-nearest neighbour method is a well-known density clustering approach [28]. Density clustering is only one of many agglomerative (or hierarchical) clustering methodologies that exist. The details of these are available in many texts [35, 36, 50, 60].

Most predictive modelling techniques may be used, to some extent, for supervised segmentation. Decision trees, which originate from classification and regression trees (CART) by Breiman *et al.* [14], are one of the most common supervised learning techniques used for model segmentation. It belongs to a subset of predictive modelling (or supervised learning) techniques called non-parametric techniques. These techniques have the useful attribute of requiring almost no assumptions about the underlying data. Decision trees use recursive partitioning algorithms to classify observations into homogenous groups, where the groups are formed through repeated attempts at finding the best possible split on the previous branch. Decision trees are relevant in various fields, like statistics [14], artificial intelligence [43] as well as machine learning [41]. Although CART is the most popular method applied in decision trees, another popular methodology for splitting is the CHAID (chi-squared automatic interaction detection) methodology [31]. CART decision trees usually do binary splits, whilst CHAID decision trees can be split into more than two nodes.

The goal of semi-supervised clustering is to guide the unsupervised clustering algorithm in finding the correct clusters by providing pairwise constraints for class labels on observations

where cluster relationships are known. The label or target here refers to a known cluster or segment label and not another target that is used for predictive modelling. Some well-known references to semi-supervised clustering include e.g., Bair [6], Basu *et al.* [7], Bilenko [11], Cohn *et al.* [18], Grira *et al.* [23], Klein *et al.* [33], and Xing *et al.* [62].

On a high level, semi-supervised clustering is performed using one of two approaches. The first approach is referred to as similarity adapting [18, 33, 62]. The second is a search-based approach [7]. Bilenko [11] describes a semi-supervised clustering method that is a combination of both these two methods (similarity adapting and search based).

Supervised clustering was formally introduced by Eick *et al.* [20]. They define the goal of supervised clustering as the quest to find "class uniform" clusters with high probability densities, otherwise known as label purity. They contrast supervised clustering with semi-supervised clustering in that all observations are "labelled", or have a target variable assigned. This is opposed to semi-supervised clustering which typically has only a small number of labelled instances. The literature on supervised clustering is quite vast [20, 27, 40, 48, 56, 63].

Note that the term semi-supervised segmentation is also found in the fields of computer vision and pattern recognition. These applications attempt to assist with identifying or grouping spatial images or objects based on their perceivable content. The principles used are similar to semi-supervised clustering, as described above, but for the purposes of segmenting photo images [29, 49, 51, 59]. For video applications see *e.g.* Badrinarayanan *et al.* [5], for ultrasound images see *e.g.* Ciurte *et al.* [17], for spine images see *e.g.* Haq *et al.* [26] and for peptide mass segmentation used in fingerprint identification see *e.g.* Bruand *et al.* [15].

The algorithm proposed in this paper for performing semi-supervised segmentation is based on k-means clustering and it applies information value [34] for the purpose of informing the segmentation decisions. The first ideas of this approach are documented in the conference paper by Breed *et al.* [13].

The abbreviation SSSKMIV is used in the remainder of this paper to refer to the proposed algorithm. The ultimate goal of the SSSKMIV algorithm (semi-supervised segmentation), as opposed to supervised or semi-supervised clustering, is not final object classification, but rather an informed separation of observations into groups on which supervised classification (or predictive modelling) can be performed, *i.e.* segmentation for predictive modelling.

3 Notation

In the proposed SSSKMIV algorithm, k-means clustering is used as the unsupervised element and information value (IV) as the supervised element. Details of the k-means clustering technique are provided below, followed by a formal definition of IV.

Consider a data set with n observations and m characteristics and let $\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ denote the i-th observation in the data set. The $n \times m$ matrix comprising all characteristics for all observations is denoted by $\mathbf{X}$. Let $\mathbf{X}_p = \{X_{1p}, X_{2p}, \ldots, X_{np}\}$ denote a vector

of all observations for a specific characteristic p.

On completion of the k-means clustering algorithm each observation $\mathbf{x}_i$, with $i = \{1, 2, \ldots, n\}$, will be assigned to one of the segments $S_1, S_2, \ldots, S_K$ where each S_j denotes an index set containing the observation indices of all the variables assigned to it. That is, if observation $\mathbf{x}_i$ is assigned to segment S_j, then $i \in S_j$. Furthermore, let $\mathbf{u}_j = \{u_{j1}, u_{j2}, \ldots, u_{jm}\}$ denote the mean (centroid) of segment S_j, for example u_{j1} will be the mean of characteristic $\mathbf{X}_1$. The distance from each observation $\mathbf{x}_i$ to the segment mean $\mathbf{u}_j$ is given by a distance function $d(\mathbf{x}_i, \mathbf{u}_j)$. If an Euclidian distance measure is used, then $d(\mathbf{x}_i, \mathbf{u}_j) = ||\mathbf{x}_i - \mathbf{u}_j||^2$ where $||\cdot||^2$ defines the length measured in Euclidean distance. The objective of the ordinary k-means clustering algorithm is to make segment assignments in order to minimise the inter-segment distances. For notational purposes $c \in \mathbb{C}$ is introduced as an index of an assignment of all the observations to different segments with $\mathbb{C}$ the set of all combinations of possible assignments. The notation S_{cj} is now introduced to reference all the observations for a given assignment $c \in \mathbb{C}$ and for a given segment index j. In addition, $\mathbf{u}_{cj}$ is the centroid of segment S_{cj}. The objective function of the ordinary k-means clustering algorithm can now be stated in generic form as

$$\min_{c \in \mathbb{C}} \sum_{j=1}^{K} \sum_{i \in S_j} d(\mathbf{x}_i, \mathbf{u}_{cj}) \tag{1}$$

For the proposed SSSKMIV algorithm, a function is required for the purpose of informing the segmentation process as part of the k-means clustering process. An example of such a function is the IV of a specific population split [34]. The IV is a measure of the separation or impurity of the target variable between segments, if the target variable is binary. Let $\mathbf{y}$ denote the vector of known target values y_i, with $i = \{1, 2, \ldots, n\}$. Consider a specific segment assignment $c \in \mathbb{C}$ and let P_{cj}^T be the proportion of events $(y_i = 1)$ of segment S_{cj} relative to the total population. Let P_{cj}^F be the proportion of non-events $(y_i = 0)$ of segment S_{cj} relative to the total population. The IV for the segment assignment $c \in \mathbb{C}$ is defined as

$$\varphi(c) = \begin{cases} \sum_{j=1}^{K} \left[\left(P_{cj}^T - P_{cj}^F \right) \times \ln \left(\frac{P_{cj}^T}{P_{cj}^F} \right) \right], & \text{if } 0 < P_{cj}^T < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In this study, the pseudo-F statistic by [16], also known as the CH measure, is used to measure the success of unsupervised segmentation. The pseudo-F statistic is not linked to any specific clustering criteria and is well suited for the purpose of measuring the success of the "unsupervised" element of the SSSKMIV algorithm. The CH measure is defined as the ratio of the separation (or "between cluster sum-of-squares") to the cohesion (or "within cluster sum-of-squares"), more specifically

$$CH = \frac{\sum_{j=1}^{K} |S_{cj}| (\mathbf{u} - \mathbf{u}_{cj})}{\sum_{j=1}^{K} \sum_{i \in S_{cj}} (\mathbf{x}_i - \mathbf{u}_{cj})} \tag{3}$$

where $\mathbf{u}$ is the mean, or centroid of the entire data set.

4 The semi-supervised segmentation algorithm (SSSKMIV)

The SSSKMIV algorithm takes two aspects into account: first, the algorithm incorporates the independent variable distribution, similarly to the k-means algorithm. Second, it focusses on target separation using a supervised function that measures the separation of the target variable between segments.

Let $0 \leq w \leq 1$ be a weight of how much the objective function of the clustering algorithm is penalised by the function that informs the segmentation process (*i.e.* the supervised weight). The proposed optimisation problem for the SSSKMIV algorithm, taking inter-segment distances into account, is the following

$$\min_{c \in \mathbb{C}} \left[w\varphi(c) + (1-w) \sum_{j=1}^{K} \sum_{i \in S_{cj}} d(\mathbf{x}_i, \mathbf{u}_{cj}) \right]. \tag{4}$$

In this paper, a heuristic approach is followed for the purpose of generating solutions to the optimisation problem in objective function (4). This includes determining the optimal supervised weight w. The algorithm consists broadly of ten steps. More details will be provided on these steps throughout the rest of this section.

In any general unsupervised k-means approach, the iterative process is terminated when no changes in the current segment assignment are made from one step to the next. Even though this could be interpreted as complete convergence, many studies have shown that the k-means algorithm is still susceptible to local optima and could arrive at different segments depending on the starting coordinates [38, 53, 54]. Due to the additional information being utilized in the SSSKMIV approach (with the order in which specific points are considered playing a role in segment assignment) complete convergence (with no observations being re-assigned) is very unlikely. For this reason, different termination criteria need to be considered which will be discussed in more detail in Step 8 below. In addition, to be in line with other studies [53], repeated runs of the algorithm are performed in order to increase the odds of finding a globally optimal solution. Each of the ten steps will now be described in more detail.

Step 1: Variable identification

Since the algorithm is based on the k-means clustering algorithm, it assumes that input variables are numerical and continuous. Furthermore, due to its isotropic nature (i.e. its tendency to form clusters that are spherical in m dimensions), it is common practice to standardise (i.e. transform to have zero mean and unit variance) all input variables for k-means analysis [37, 45, 54]. This also applies to the SSSKMIV algorithm.

However, in practical applications of SSSKMIV it is likely that the independent variables contain values of a categorical nature. The numerical and continuous assumption is not unique to clustering algorithms, but is also present in regression analysis. There are numerous techniques available to convert categorical variables to numeric: *e.g.* single standardised target rates [2]; using weights of evidence [47]; optimal scores using correspondence analysis [52] and using 'dummy' variables [4]. For the purpose of this paper

we used 'dummy' variables for accommodating categorical inputs. This specific technique does not force a specific relationship with the target variable for any of the segments [3].

Step 2: Segment seed initialisation

Cluster seed initialisation has been the focus of many studies [30, 32, 39, 42]. Although the studies vary in their recommendations, it is clear that the initialisation of cluster centres has an impact on the final results. Certain techniques are able to improve the speed with which convergence is achieved, but may bias the final result [39].

Since the SSSKMIV algorithm adds an additional dimension to the standard k-means algorithm, it is expected that the initialisation techniques that are proposed for k-means above will, however, be less effective. This is due to the fact that these techniques are generally focussed on density approximations and do not take the dependent variable into account.

Two methodologies were considered for random initialisation: initialisation based on variable range [39] and initialisation based on random observation selection [32, 39]. The latter was chosen based on empirical analysis [12] which showed that this method reduces the probability of segments being initialised without any assigned observations.

Step 3: Initial data set and variable preparation

Step 3 initialises the data set with the required variables needed for the semi-supervised segmentation analysis. This is the last initialisation step before the iterative *assignment evaluation and update* (Step 5) process commences.

Step 4: Assignment

The assignment step assigns observations to segments in order to improve objective function (4) of the SSSKMIV. Each observation $\mathbf{x}_i$ is put through several sub-steps which is discussed here. First, the Euclidian distances between the observation and all segment centres are calculated. The output is a vector $\mathbf{d} = \{d_1, d_2, \ldots, d_K\}$ that contains the Euclidian distance $d(\mathbf{x}_i, \mathbf{u}_{cj})$ for each segment S_{cj} where $j = \{1, 2, \ldots, K\}$. Second, the output vector φ (referred to as the supervised values) is calculated based on a given assignment of the observation i to each of the segments. It should be noted that if the supervised function returns zero (see equation (2)), then the segment allocation will be made based solely on the Euclidean distance $d(\mathbf{x}_i, \mathbf{u}_{cj})$ (see equation (4)).

The third sub-step is to standardise the distances and supervised values. For each observation i, the distance d_j and supervised factor φ_j is respectively replaced with standardised distance d'_j and standardised supervised factor φ'_j for every segment S_{cj} by subtracting the average and dividing by the standard deviation. The fourth sub-step is to assign each observation i in such a way as to minimise the value of the objective function. More specifically, assign observation i to $S_{cj_{min}}$ where $j_{min} = \arg\min_{j=1,\ldots,K}[w\varphi'_j + (1-w)d'_j]$. This equation is referred to as the local objective function of the SSSKMIV.

Step 5: Assignment evaluation and update

Similar to the standard k-means algorithm, the update step of the SSSKMIV algorithm updates the segment centroids based on the new assignments made in Step 4. This step also evaluates the assignments made to ensure all segments have observations assigned to it. Since the algorithm assumes a pre-specified number of segments, the segmentation process will be randomly re-initiated if a segment has not been assigned any observations.

Step 6: Summarise and log step statistics

In order to assess various aspects and results of the SSSKMIV, a number of key statistics are logged throughout the process. These are the coordinates of the segment centres after each iteration; the distance moved by each centroid after each iteration; the target rate (or target average) of each segment after each iteration; the percentage of the observations in each segment after each iteration; the CH value of the segmentation after each iteration; the value of the supervised function after each iteration; the relative distance of each segment to the other segments; the number of segment assignments that was changed due to the influence of the supervised factor (*i.e.* how many observations were assigned to a different segment due to the addition of the supervised factor to the objective function); and finally the time and speed with which each iteration was performed as well as an estimated termination time as calculated after each iteration.

Step 7: Randomise data set

As explained earlier, the order in which observations are assessed could make a difference to the segment they are assigned to. In order to avoid the order in which observations are assessed biasing the final output, the observations are randomly resorted after each assignment step. This biasing effect was pointed out by Wagstaff *et al.* [58] and a number of subsequent studies in supervised [20] and semi-supervised clustering [8, 11] implemented similar measures to avoid it.

Step 8: Evaluate stopping criterion

In standard k-means analyses, the iterative assignment and update process are stopped when no assignments change from one step to the next. This works well for k-means clustering and may be considered as a sufficient convergence criterion. This is however very unlikely in the case of SSSKMIV as applied in this study. This is once again due to the supervised function being dependent on the order in which observations are assessed. Similar behaviour was observed in other studies regarding supervised clustering [40]. For this reason the stopping criterion needed to be reconsidered for the SSSKMIV algorithm. The following basic stopping criterion was followed: First if the standard k-means clustering convergence criterion is assessed, and if no observations were reassigned after the previous step, the process is stopped. Else, if the standard convergence criterion is not met, the average distance that the segment centroids moved from one step to the next is measured for a number of runs. As long as the average distance that the segment centroids travel is still decreasing, the process is repeated. Whenever the average distance increases from

one step to the next the process is terminated. More complex stopping criteria may be used, but this may, however, be detrimental to computing times.

Step 9: Over fit evaluation and smoothing

As with most supervised classification algorithms, the SSSKMIV algorithm may over fit when values become too large. This happens when observations are no longer logically assigned to segments based on independent variable proximity, but almost entirely due to their target value. When this happens, segments can no longer be applied on new data sets, since it cannot be described through their independent variables. As a counter measure to prevent over fitting, the SSSKMIV applies k-nearest neighbourhood (KNN) smoothing. This methodology is also used to assign segments to the validation set by using the development set as input. The KNN smoothing methodology is well-established and more detail can be found in the literature [22].

Step 10: Final evaluation and result logging

After applying the nine steps described above on the development data set, the results obtained for the validation dataset can be evaluated. As part of this step in the algorithm, further statistics on both the development and validation data sets are computed, so that the result can be compared to other runs. The statistics that are calculated are: the new segment centroids after the smoothing exercise for both the validation and development set; the target rate for each segment; the final population percentage in each segment; the CH value to describe the quality of the segmentation from an independent variable perspective; the overall supervised value (*i.e.* the IV value); the final Euclidian distances between segment centroids and the impact of the smoothing exercise which is expressed as a percentage of the validation set's observations that remained the same. The data set can then be used for development of statistical models and the impact measured on the validation data set.

5 Simulation study results

In this section, the performance of the SSSKMIV algorithm is demonstrated by comparing its results to the results obtained by both supervised and unsupervised techniques. Decision trees are employed as the supervised technique and k-means clustering as the unsupervised technique. In order to analyse the performance of the three different segmentation approaches, simulated data with predefined characteristics were used. This may help to understand what characteristics cause one methodology to outperform another. It should be noted that all possible data characteristics are not simulated here (that would be impossible), but simply some of the more obvious ones.

In order to facilitate a good platform to explain the data simulation experiment, we first establish a base case for simulating a data set, after which the additional elements that are varied for further exploration are discussed. The approach described here is similar to approaches of simulating data sets with binary outcomes followed by Shifa & Rashid [46] and Venter & De la Rey [57].

	$\mathbf{X}_3$		$\mathbf{X}_4$	
Segment	**Mean**	**Variance**	**Mean**	**Variance**
S_1	4	1	6	2
S_2	10	2	12	1
S_3	−4	2	−2	1
S_4	−2	1	−4	2
S_5	1	5	10	5
S_6	5	5	5	5

Table 1: The different normal distributions used for each segment.

The goal of this base case scenario is to show that it is possible to simulate a data set on which segmentation for logistic regression modelling will have a positive impact on accuracy, compared to the case where no segmentation is done.

In the base case scenario, the number of segments is assumed to be six ($K = 6$) and the number of characteristics is assumed to be twenty ($m = 20$). Different weights for β are used for each of the six segments. For S_1 to S_6, the first four values of β, and all other values of β will be zero, *i.e.* $\beta_7, \ldots, \beta_{20}$ will be set to zero. For S_2 the values of $\beta_7 = -1$ and $\beta_8 = 1$ and all other values of β will be zero. This pattern will continue until the sixth segment, *i.e.* for S_6 the values of $\beta_{15} = -1$ and $\beta_{16} = 1$ and all other values of β will be zero. In all cases, β_0 is set to 0.

All values in $\mathbf{X}$, except for $\mathbf{X}_3$ and $\mathbf{X}_4$, are drawn from $N(0, 1)$ distribution. In order to distinguish the segments, $\mathbf{X}_3$ and $\mathbf{X}_4$ were drawn from separate normal distributions for each segment as indicated in Table 1.

The number of observations per segment were also varied as follows: $S_1 : 1\,000$, $S_2 : 200$, $S_3 : 500$, $S_4 : 1\,000$, $S_5 : 1\,000$ and $S_6 : 500$. The resulting probability vectors, associated with each of the six segments, are:

$$\mathbf{p}_1 = \frac{1}{1 + e^{-(\mathbf{X}_1 + \mathbf{X}_2 - 0.5\mathbf{X}_3 + 0.5\mathbf{X}_4 - \mathbf{X}_5 + \mathbf{X}_6)}},$$
$$\mathbf{p}_2 = \frac{1}{1 + e^{-(\mathbf{X}_1 + \mathbf{X}_2 - 0.5\mathbf{X}_3 + 0.5\mathbf{X}_4 - \mathbf{X}_7 + \mathbf{X}_8)}},$$
$$\mathbf{p}_3 = \frac{1}{1 + e^{-(\mathbf{X}_1 + \mathbf{X}_2 - 0.5\mathbf{X}_3 + 0.5\mathbf{X}_4 - \mathbf{X}_9 + \mathbf{X}_{10})}},$$
$$\mathbf{p}_4 = \frac{1}{1 + e^{-(\mathbf{X}_1 + \mathbf{X}_2 - 0.5\mathbf{X}_3 + 0.5\mathbf{X}_4 - \mathbf{X}_{11} + \mathbf{X}_{12})}},$$
$$\mathbf{p}_5 = \frac{1}{1 + e^{-(\mathbf{X}_1 + \mathbf{X}_2 - 0.5\mathbf{X}_3 + 0.5\mathbf{X}_4 - \mathbf{X}_{13} + \mathbf{X}_{14})}},$$
$$\mathbf{p}_6 = \frac{1}{1 + e^{-(\mathbf{X}_1 + \mathbf{X}_2 - 0.5\mathbf{X}_3 + 0.5\mathbf{X}_4 - \mathbf{X}_{15} + \mathbf{X}_{16})}}.$$

Since $\mathbf{y}$ is binary, assign $\mathbf{y}$ as

$$y_i = \begin{cases} 1, & u_i \leq p_i \\ 0, & u_i > p_i, \end{cases} \tag{5}$$

where $\mathbf{u} = \{u_1, \ldots, u_n\}$ and the elements of $\mathbf{u}$ are independently drawn from a $U(0, 1)$

distribution, and p_i is the component of the probability vector $\mathbf{p}_j$ corresponding to observation $\mathbf{x}_i$.

A total of 100 datasets were generated as described above. Note that the event rate in each segment will differ (due the way the simulation was structured). The average event rate for the 100 simulated datasets is around 35% for Segment 4 and 90% for Segment 5. The average IV value for the segments is 0.9 and the average CH value is 0.27.

The data were divided into a development and a validation set. The development set is used to perform the segmentation on and for developing the predictive models with, whilst the validation set is used to test the lift in model accuracy (as measured by the Gini coefficient). The development and validation sets were generally sampled with equal sizes (*i.e.* 50% each). There are different ways to calculate the Gini coefficient, as well as different names for this statistic, *e.g.* accuracy ratio (defined as the summary statistic of the cumulative accuracy profile), the Somers D statistic (defined as the ratio of the concordant and discordant pairs as a ratio of all possible pairs), and the Gini coefficient is also closely related to the area under the receiving operating curve, *i.e.* two times the Gini coefficient less one, is equal to the area under the receiving operating curve [2, 47, 55].

A single logistic regression model was fitted to the entire development set (*i.e.* the unsegmented dataset). To this end stepwise regression was applied using SAS software's Proc Logistic [44]. The significance level for entry of parameters was set at 0.1, whilst the significance level for removal was set at 0.05. The resulting model provides the reference model against which the segmented models' accuracy will be tested by calculating the Gini coefficient.

The development set was also split into the different segments (using three different techniques of segmentation), on which separate logistic models were developed (using the same settings as described above). Once all models have been developed, they were applied to the validation set. The unsegmented model was applied to the full validation set to obtain the reference Gini coefficient, whilst the segmented models were applied individually to each corresponding segment. In order to measure the combined Gini coefficient of the segmented models on the validation set, the predicted probabilities of all segments were combined, and the Gini coefficient was calculated on the overall, combined set. Once this is done, the unsegmented Gini coefficient can be compared to the combined, segmented Gini coefficient.

The best validation reference Gini coefficient (*i.e.* die Gini coefficient on the unsegmented data) was 71.8%. By using the known six segments, and fitted six logistic regressions to these six segments, the best validation Gini coefficient was 81.9%. It is evident that by using perfect segmentation, it is possible to improve the Gini coefficient by 10% (from 71.8% to 81.9%).

Supervised segmentation was performed by means of a decision tree and Proc Split in SAS was used to segment the data sets. Since the goal is to develop predictive models (which cannot be done effectively on very small samples), the "Splitsize" option was used to set the minimum number of observations in a leaf and control the number of segments created. The procedure will still consider other options for splitting the node, but will simply eliminate those that result in leaves that will breach the indicated "Splitsize" value.

Method	Best validation set Gini	Best Gini improvement (over 71.8%)	Average IV	Average CH value
Unsupervised segmentation (using k-means)	74.45%	2.46%	0.35	0.090
Semi-supervised segmentation (using SSKMIV)	74.89%	2.90%	0.64	0.087
Supervised segmentation (using decision trees)	73.42%	1.43%	1.01	0.069

Table 2: A summary of the success of different segmentation algorithms.

The initial value used was the number of observations in the development set divided by two times K (where K is the selected number of segments).

For the purpose of performing unsupervised segmentation, k-means clustering was applied by simply using the SSKMIV algorithm and choosing $w = 0$ (*i.e.* only the unsupervised element was taken into account). For the purpose of performing semi-supervised segmentation, the SSSKMIV was applied, while considering the supervised weight values $w \in \{0.1, 0.2, 0.3, \ldots, 0.7, 0.8, 0.9\}$.

The results obtained when applying the three segmentation techniques to the generated data are summarised in Table 2. From the results it is observed that the SSKMIV algorithm outperforms both the unsupervised and supervised approaches. The unsupervised segmentation forms segments with the highest CH values, but the lowest IV values, whilst the supervised segmentation forms segments that obtain the highest IV and the lowest CH value. The semi-supervised segmentation strikes a good balance between the two, but can only achieve a 2.9% Gini coefficient improvement at best (this is achieved with $w = 0.7$). This is significantly lower than the optimal improvement of more than 10% (if we had perfect knowledge on the segments).

Some characteristics of the data on which segmentation for predictive modelling is performed can be controlled by simulating data sets. This provides the opportunity to explore links between data set characteristics and dominance of a specific segmentation technique. Practical data sets are in most cases made up of real-world data, which are extremely complex and diverse, making it unreasonable to find an exhaustive list of reasons for one technique outperforming another. In an attempt to explore some of the more obvious links, the impact of varying three main characteristics in the simulated data sets was explored. For this purpose target rate separation between segments was controlled, as measured by the IV. Secondly, the difference in the independent variable distribution was controlled, as measured by CH value. Thirdly, the segment complexity, defined as $\mathcal{O}$, was controlled, as measured by the number of independent variables that was used to define a segment. Again the combined Gini coefficient improvement of the segmented models was compared with the Gini coefficient obtained with no segmentation.

A similar approach for performing the additional simulations was followed as described above. A few additional steps were, however, performed in order to ensure that the IV and CH values differ. Each segment size (SS_j) was drawn from a normal distribution such that $SS_j \sim N(\bar{SS}, 0.2 \times \bar{SS})$ where $\bar{SS}$ is the average size of the segment (chosen as 1 000).

The event rate was drawn from a normal distribution $N(0.5, 0.2)$ such that each event rate is between 0.02 and 0.98. The limits of 2% and 98% were set to ensure an IV can always be calculated for each simulated data set, since IVs are not defined for bad rates of 100% or 0% [47]. The segment complexity, $\mathcal{O}$, is the number of independent variables that was used to define a segment. For each of these variables used to define a segment, the mean and the standard deviation were drawn from the uniform distribution $U(0, 3)$. Variables $\mathbf{X}_1$ to $\mathbf{X}_{11}$ were generated as described above, but variables $\mathbf{X}_{12}$ to $\mathbf{X}_{26}$ were subjected to variation depending on the complexity selected. For this purpose the parameter $\mathcal{O}$ was allowed to be varied between 1 and 15. More specifically, if $\mathcal{O} = 15$ a total of 26 variables were chosen.

A total of 20 000 simulated datasets were generated while considering complexity values of $\mathcal{O} \in \{5, 10, 15\}$. The IV values were grouped into four groups namely, $(0, 0.05]$, $(0.05, 0.5]$, $(0.5, 0.8]$ and above 0.8. The values used for the ranges of each of the groups were based on analysis of the results observed for the different segmentation techniques on data sets with IVs in these ranges. As will be seen in the results section, grouping the IVs in this way provides us with enough volume in IV areas where different segmentation techniques perform well. This provides a comparative view of how IVs can influence the effectiveness of specific segmentation techniques.

The allowable range of CH values differ depending on the value of the complexity parameter $\mathcal{O}$. For this purpose, all the scenarios that were generated for a specific value of $\mathcal{O}$ were divided into deciles (ranked groups consisting of 10% of the total number of scenarios) based on the CH value. Only scenarios from the first, fifth and tenth decile for a specific value of $\mathcal{O}$ were selected for the purpose of obtaining a good spread of CH values without the need to do too many iterations. The first decile contains the highest CH values, and is therefore called the "High" group. The fifth decile contains mid-range values of the CH value, and is therefore called "Mid". The tenth decile contains the lowest CH values, and is subsequently called "Low".

K-means clustering was applied again as the unsupervised segmentation technique, decision trees as the supervised segmentation technique and SSKMIV as the semi-supervised segmentation technique, while using $w \in \{0.25, 0.5, 0.75\}$. The selections made above meant that a total number of 1 800 segmentation iterations was performed and 10 800 models developed for each value of $\mathcal{O}$. In addition to this, time was required to select and generate the required data sets. Even though the size of the data sets were relatively small, the estimated time required to perform the analyses per value of $\mathcal{O}$ was between four and five days.

The discussion to follow contrasts the lowest complexity case ($\mathcal{O} = 0.5$) with the highest ($\mathcal{O} = 15$) since by doing this, the results obtained for the scenario where ($\mathcal{O} = 10$), are more clearly put into perspective. Table 3 summarizes the results for $\mathcal{O} = 5$. The best possible Gini coefficient improvement percentage was obtained by the supervised segmentation (decision trees) when CH values are high and IVs are greater than 0.8. This group obtains an average of just over 40% of the true Gini coefficient improvement. Although the decision tree clearly dominates for the most part, the stable performance of the SSSKMIV algorithm is clear, since the SSSKMIV algorithm shows improvement over non-segmented models in all but one group. This is not the case with unsupervised

Data		Gini % improvement		
IV	**CH**	**Unsupervised segmentation**	**Semi-supervised segmentation**	**Supervised segmentation**
(0, 0.05]	Low	−6.79%	−1.79%	−11.08%
(0, 0.05]	Mid	−2.06%	1.75%	−8.85%
(0, 0.05]	High	4.16%	9.53%	−7.20%
(0.05, 0.5]	Low	−6.61%	0.10%	5.31%
(0.05, 0.5]	Mid	−0.56%	5.28%	7.62%
(0.05, 0.5]	High	4.62%	11.38%	19.56%
(0.5, 0.8]	Low	−1.10%	3.79%	19.09%
(0.5, 0.8]	Mid	−0.11%	8.40%	27.59%
(0.5, 0.8]	High	9.79%	18.84%	32.08%
Above 0.8	Low	−4.87%	5.67%	19.49%
Above 0.8	Mid	7.38%	18.63%	35.98%
Above 0.8	High	13.54%	25.48%	41.15%

Table 3: Improvement in Gini by CH and IV group ($\mathcal{O} = 5$)

segmentation (k-means) or supervised segmentation (decision trees). This is indicated by the negative signs in seven of the twelve cases for the unsupervised segmentation (up to 6.79% worse than the non-segmented models) and two negative signs in the twelve cases for the supervised segmentation (up to 11.08% worse than the non-segmented models). The semi-supervised segmentation (SSSKMIV) only had one negative sign while only being 1.79% worse than the non-segmented models.

Table 4 summarize the results for $\mathcal{O} = 15$. The best performance overall in this case is the SSSKMIV algorithm. The highest improvement achieved is 73.99% of the true Gini coefficient improvement on average in the high CH value, high IV group by the SSSKMIV. The IV group between 0.0 and 0.05, with high CH values, is most closely contested, with the unsupervised k-means algorithm (52.43%) obtaining results very close to the SSSKMIV algorithm (53.35%). When the complexity is high, the semi-supervised segmentation (SSSKMIV) outperforms both the supervised and the unsupervised segmentation.

Lastly the middle group, where $\mathcal{O} = 10$ is considered in Table 5. In nine of the twelve cases, the semi-supervised segmentation (SSSMIV) outperforms the supervised segmentation (decision trees). In this specific analysis, the unsupervised segmentation never outperforms the semi-supervised segmentation, although it does outperform the supervised segmentation in five cases (especially in the low IV groups).

A note on the value of w to use in SSSMIV: there is no single value of w that always outperforms independent of the data set characteristics. It is, therefore, not possible to recommend a good value of w. The most appropriate value of w is to be determined iteratively on every data set that is segmented using the SSSKMIV algorithm.

Data		Gini % improvement		
IV	**CH**	**Unsupervised segmentation**	**Semi-supervised segmentation**	**Supervised segmentation**
$(0, 0.05]$	Low	28.22%	29.91%	−7.09%
$(0, 0.05]$	Mid	38.37%	41.94%	−5.05%
$(0, 0.05]$	High	52.43%	53.35%	−2.67%
$(0.05, 0.5]$	Low	27.91%	36.29%	16.99%
$(0.05, 0.5]$	Mid	42.88%	50.61%	17.22%
$(0.05, 0.5]$	High	55.29%	62.55%	25.32%
$(0.5, 0.8]$	Low	31.43%	43.86%	29.29%
$(0.5, 0.8]$	Mid	47.73%	57.78%	33.43%
$(0.5, 0.8]$	High	55.58%	68.16%	41.20%
Above 0.8	Low	43.94%	55.10%	42.46%
Above 0.8	Mid	51.63%	64.99%	50.14%
Above 0.8	High	62.96%	73.99%	54.01%

Table 4: Improvement in Gini by CH and IV group ($\mathcal{O} = 15$)

Data		Gini % improvement		
IV	**CH**	**Unsupervised segmentation**	**Semi-supervised segmentation**	**Supervised segmentation**
$(0, 0.05]$	Low	5.52%	9.97%	−9.29%
$(0, 0.05]$	Mid	17.70%	21.26%	−4.24%
$(0, 0.05]$	High	29.66%	32.47%	−4.66%
$(0.05, 0.5]$	Low	8.85%	16.80%	11.78%
$(0.05, 0.5]$	Mid	20.38%	27.99%	20.45%
$(0.05, 0.5]$	High	29.52%	37.88%	18.68%
$(0.5, 0.8]$	Low	14.09%	23.54%	29.21%
$(0.5, 0.8]$	Mid	25.37%	34.42%	33.78%
$(0.5, 0.8]$	High	40.11%	48.83%	40.12%
Above 0.8	Low	18.37%	31.54%	37.27%
Above 0.8	Mid	27.94%	41.58%	44.30%
Above 0.8	High	43.49%	53.46%	49.86%

Table 5: Improvement in Gini by CH and IV group ($\mathcal{O} = 10$)

6 Conclusions and future research

The objective function of the semi-supervised algorithm (SSSKMIV) proposed in this paper, comprises a supervised element (using information value) and an unsupervised element (using k-means clustering). In addition, a supervised weight w was introduced which measures how much the objective function of the unsupervised element is penalised by the supervised element.

Empirical tests were performed on simulated data to demonstrate the performance of the proposed SSSKMIV, compared to a supervised and an unsupervised segmentation approach. It was found that data sets which comprise complex underlying segments, and are described by many variables, are not well suited to supervised segmentation. The segmentation will be more successful if unsupervised or semi-supervised techniques are used. Conversely, supervised segmentation appears to be more successful when segments

are simple (*i.e.* not described by many independent variables), and target rates differ substantially between segments. The SSSKMIV algorithm consistently performed well within a large range of data set characteristics, and outperformed known techniques in many instances. Specifically, when the complexity is high, the semi-supervised segmentation (SSSKMIV) outperforms both the supervised and unsupervised segmentation.

Within this study, not all avenues of possible research could be explored, and some are therefore left for future work. One aspect identified for future research is the efficiency and execution time of the proposed semi-supervised segmentation algorithm. Another challenge is to determine a narrower band for the supervised weight (w), which could also be useful in reducing the number of required iterations. In this study, only data sets with binary target variables were considered, due to the use of IV as a supervised function. Further studies could focus on extending the algorithm proposed in this paper to models with continuous target variables.

Acknowledgement

The authors gratefully acknowledge the valuable input given by the anonymous referees. This work is based on research supported in part by the Department of Science and Technology (DST) of South Africa. The grant holder acknowledges that opinions, findings and conclusions or recommendations expressed in any publication generated by DST-supported research are those of the author(s) and that the DST accepts no liability whatsoever in this regard.

References

[1] Anderberg MR, 1973, *Cluster analysis for applications*, Technical Report: DTIC, [Online], Available from http://www.dtic.mil/docs/citations/AD0770256.

[2] Anderson R, 2007, *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation*, 1st Edition, Oxford University Press, New York (NY).

[3] Anderson R, 2015, *Piecewise Logistic Regression: an Application in Credit Scoring*, Credit Scoring and Control Conference XIV, August 26–28, 2015, Edinburgh.

[4] Anttila-Hughes JK & Hsiang SM, 2013, *Destruction, disinvestment, and death: Economic and human losses following environmental disaster*, Working Paper, Available at SSRN 2220501.

[5] Badrinarayanan V, Budvytis I & Cipolla R, 2013, *Semi-supervised video segmentation using tree structured graphical models*, Pattern Analysis and Machine Intelligence, **35(11)**, pp. 2751–2764.

[6] Bair E, 2013, *Semi-supervised clustering methods*, Wiley Interdisciplinary Reviews: Computational Statistics, **5(5)**, pp. 349–361.

[7] Basu S, Banerjee A & Mooney R, 2002, *Semi-supervised clustering by seeding*, Proceedings of the 19th International Conference on Machine Learning (ICML-2002), 8–12 July 2002, Sydney.

[8] Basu S, Bilenko M & Mooney R, 2004, *A probabilistic framework for semi-supervised clustering*, Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 22–25 August 2004, Seattle (WA), pp. 59–68.

[9] Baum LE & Petrie T, 1966, *Statistical inference for probabilistic functions of finite state Markov chains*, The annals of mathematical statistics, **37(6)**, pp. 1554–1563.

[10] Bijnen E, Stouthard P & Brand-Maher C, 2012, *Cluster analysis: Survey and evaluation of techniques*, Tilburg University Press, Tilburg.

[11] Bilenko M, 2004, *Integrating constraints and metric learning in semi-supervised clustering*, Proceedings of the ICML, 4–8 July 2004, Alberta.

[12] Breed D, 2017, *Semi-supervised segmentation within a predictive modelling context*, Doctoral Dissertation, North-West University, Potchefstroom Campus.

[13] Breed D, De La Rey T & Terblanche S, 2013, *The use of different clustering algorithms and distortion functions in semi supervised segmentation*, Proceedings of the 42nd ORSSA Annual Conference, 15–18 September 2013, Stellenbosch, pp. 122–132.

[14] Breiman L, Fredman J, Olsen R & Stone C, 1984, *Classification and Regression Trees*, Wadsworth, Pacific Grove (CA).

[15] Bruand J, Alexandrov T, Sistla S, Wisztorski M, Meriaux C, Becker M, Salzet M, Fournier I, Macagno E & Bafna V, 2011, *AMASS: algorithm for MSI analysis by semi-supervised segmentation*, Journal of Proteome Research, **10(10)**, pp. 4734–4743.

[16] Caliński T & Harabasz J, 1974, *A dendrite method for cluster analysis*, Communications in Statistics-theory and Methods, **3(1)**, pp. 1–27.

[17] Ciurte A, Bresson X, Cuisenaire O, Houhou N, Nedevschi S, Thiran JP & Cuadra MB, 2014, *Semi-supervised segmentation of ultrasound images based on patch representation and continuous min cut*, PloS one, **9(7)**, pp. 1–14.

[18] Cohn D, Caruana R & Mccallum A, 2003, *Semi-supervised Clustering with User Feedback*, Technical Report: Cornell University.

[19] Cross G, 2008, *Understanding your customer: segmentation techniques for gaining customer insight and predicting risk in the telecom industry*, Paper 154-2008 , SAS Global Forum 2008.

[20] Eick C, Zeidat N & Zhao Z, 2004, *Supervised Clustering Algorithms and Benefits*, Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI04, 15–17 November 2004, Boca Raton (FL), pp. 774–776.

[21] FICO, 2014, *Using segmented models for better decisions*, Technical Report: Fair Isaacs, [Online], Available from http://www.fico.com/en/node/8140?file=9737.

[22] Friedman J, Hastie T & Tibshirani R, 2001, *The elements of statistical learning*, Springer Series in Statistics, Springer, Berlin.

[23] Grira N, Crucianu M & Boujemaa N, 2005, *Unsupervised and Semi-supervised Clustering: a Brief Survey*, A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6), [Online], Available from http://cedric.cnam.fr/~crucianm/src/BriefSurveyClustering.pdf.

[24] Hand D, 1997, *Construction and Assessment of Classification Rules*, John Wiley & Sons Ltd., Chichester, England.

[25] Hand D, 2005, *What you get is what you want? some dangers of black box data mining*, Proceedings of the M2005 Conference, SAS Institute, 24–25 October 2005, Las Vegas (NV).

[26] Haq R, Aras R, Besachio DA, Borgie RC & Audette MA, 2015, *Minimally Supervised Segmentation and Meshing of 3D Intervertebral Discs of the Lumbar Spine for Discectomy Simulation*, pp. 143–155 in *Recent Advances in Computational Methods and Clinical Applications for Spine Imaging*, pp. 143–155. Springer, Berlin.

[27] Harremoës P & Tishby N, 2007, *The information bottleneck revisited or how to choose a good distortion measure*, IEEE International Symposium on Information Theory, 2007. ISIT 2007, 24–29 June 2007, Nice, France, pp. 566–570.

[28] Hartigan J, 1997, *Clustering algorithms*, 1st Edition, John Wiley & Sons Ltd., New York (NY).

[29] Houhou N, Bresson X, Szlam A, Chan T & Thiran JP, 2009, *Semi-supervised segmentation based on non-local continuous min-cut*, Scale Space and Variational Methods in Computer Vision, **1(1)**, pp. 112–123.

[30] Kang P & Cho S, 2009, *K-means clustering seeds initialization based on centrality, sparsity, and isotropy*, International Conference on Intelligent Data Engineering and Automated Learning, **1(1)**, pp. 109–117.

[31] Kass GV, 1980, *An exploratory technique for investigating large quantities of categorical data*, Applied Statistics, **29(2)**, pp. 119–127.

[32] Khan SS & Ahmad A, 2004, *Cluster center initialization algorithm for K-means clustering*, Pattern Recognition Letters, **25(11)**, pp. 1293–1302.

[33] Klein D, Kamvar SD & Manning CD, 2002, *From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering*, Technical Report, [Online], Available from http://ilpubs.stanford.edu:8090/528/1/2002-10.pdf Stanford, San Francisco.

[34] Kullback S, 1958, *Information Theory and Statistics*, 1st Edition, John Wiley & Sons Ltd., New York (NY).

[35] Lance GN & Williams WT, 1967, *A general theory of classificatory sorting strategies I: Hierarchical Systems*, The computer journal, **9(1)**, pp. 373–380.

[36] Massart D & Kaufman L, 1983, *The interpretation of analytical chemical data by the use of cluster analysis*, Wiley: The University of Michigan, Ann Arbor (MI).

[37] Milligan GW & Cooper MC, 1988, *A study of standardization of variables in cluster analysis*, Journal of Classification, **5(2)**, pp. 181–204.

[38] Pelleg D & Moore AW, 2000, *X-means: Extending k-means with efficient estimation of the number of clusters*, Proceedings of the ICML, 29 June – 2 July 2000, Stanford (CA), pp. 727–734.

[39] Pena JM, Lozano JA & Larranaga P, 1999, *An empirical comparison of four initialization methods for the k-means algorithm*, Pattern recognition letters, **20(10)**, pp. 1027–1040.

[40] Peralta B, Espinace P & Soto A, 2013, *Enhancing k-means using class labels*, Intelligent Data Analysis, **17(6)**, pp. 1023–1039.

[41] Quinlan J, 1993, *Programs for Machine Learning*, Morgan Kaufman, San Mateo (CA).

[42] Redmond SJ & Heneghan C, 2007, *A method for initialising the k-means clustering algorithm using kd-trees*, Pattern Recognition Letters, **28(8)**, pp. 965–973.

[43] Safavian SR & Landgrebe D, 1991, *A survey of decision tree classifier methodology*, IEEE transactions on systems, man, and cybernetics, **21(3)**, pp. 660–674.

[44] SAS Institute Inc, 2015, *Applied Analytics Using SAS Enterprise Miner (SAS Institute Course Notes)*, SAS Institute Inc., Cary (NC).

[45] Schaffer CM & Green PE, 1996, *An empirical comparison of variable standardization methods in cluster analysis*, Multivariate Behavioral Research, **31(2)**, pp. 149–167.

[46] Shifa N & Rashid M, 2003, *Monte Carlo Evaluation of Consistency and Normality of Dichotomous Logistic and Multinomial Logistic Regression Models*, Hawaii University, Honolulu (HI).

[47] Siddiqi N, 2006, *Credit Risk Scorecards*, John Wiley & Sons, Hoboken (NJ).

[48] Sinkkonen J, Kaski S & Nikkilä J, 2002, *Discriminative clustering: Optimal contingency tables by learning metrics*, Proceedings of the European Conference on Machine Learning, 19-23 August 2002, Helsinki, Finland, pp. 418–430.

[49] Socher R & Fei-Fei L, 2010, *Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora*, Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 13–18 June 2010, San Francisco (CA), pp. 966–973.

[50] Sokal RR, 1958, *A statistical method for evaluating systematic relationships*, The University of Kansas science bulletin, **38(1)**, pp. 1409–1438.

[51] Song X & Fan G, 2002, *A study of supervised, semi-supervised and unsupervised multiscale bayesian image segmentation*, Proceedings of the 2002 45th Midwest Symposium on Circuits and Systems, 4–7 August 2002, Tulsa (OK), pp. 2–371.

[52] Sourial N, Wolfson C, Zhu B, Quail J, Fletcher J, Karunananthan S, Bandeen-Roche K, Béland F & Bergman H, 2010, *Correspondence analysis is a useful tool to uncover the relationships among categorical variables*, Journal of clinical epidemiology, **63(6)**, pp. 638–646.

[53] Steinley D, 2003, *Local optima in k-means clustering: what you don't know may hurt you*, Psychological Methods, **8(3)**, pp. 294–304.

[54] Steinley D, 2004, *Standardizing variables in k-means clustering*, Proceedings of the Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), 15–18 July 2004, Chicago (IL), pp. 53–60.

[55] Thomas LC, 2009, *Consumer Credit Models: Pricing, Profit and Portfolios: Pricing, Profit and Portfolios*, Oxford University Press, New York (NY).

[56] Tishby N, Pereira FC & Bialek W, 2000, *The information bottleneck method*, The 37th annual Allerton Conference on Communication, Control, and Computing, 4–6 October 2000, Champaign (IL), pp. 368–377.

[57] Venter J & De la Rey T, 2007, *Detecting outliers using weights in logistic regression*, South African Statistical Journal, **41(2)**, pp. 127–160.

[58] Wagstaff K, Cardie C, Rogers S, Schrödl S *et al.*, 2001, *Constrained k-means clustering with background knowledge*, Proceedings of the ICML, 28 June – 1 July 2001, Williamstown (MA), pp. 577–584.

[59] Wang A, Li J, Wu P & Lu Z, 2011, *Semi-supervised Segmentation Based on Level Set*, Proceedings of the Information and Business Intelligence: International Conference, IBI 2011, 23–25 December 2011, Chongqing, China, pp. 129–135.

[60] Ward Jr JH, 1963, *Hierarchical grouping to optimize an objective function*, Journal of the American statistical association, **58(301)**, pp. 236–244.

[61] Wong MA & Lane T, 1981, *A kth nearest neighbour clustering procedure*, Proceedings of the 13th Symposium on the Interface: Computer Science and Statistics, 12–13 March 1981, Pittsburgh (PA), pp. 308–311.

[62] Xing EP, Ng AY, Jordan MI & Russell S, 2002, *Distance metric learning with application to clustering with side-information*, Proceedings of the Advances in neural information processing systems, 9–14 December 2002, Vancouver, Canada, pp. 505–512.

[63] Ye J, Zhao Z & Wu M, 2008, *Discriminative k-means for clustering*, Proceedings of the Advances in neural information processing systems, 8–11 December 2008, Vancouver, Canada, pp. 1649–1656.