North American Actuarial Journal

ISSN: 1092-0277 (Print) 2325-0453 (Online) Journal homepage: www.tandfonline.com/journals/uaaj20

Data Clustering with Actuarial Applications

Guojun Gan & Emiliano A. Valdez

To cite this article: Guojun Gan & Emiliano A. Valdez (2020) Data Clustering with Actuarial Applications, North American Actuarial Journal, 24:2, 168-186, DOI: 10.1080/10920277.2019.1575242

To link to this article: https://doi.org/10.1080/10920277.2019.1575242

Published online: 14 Jun 2019.

North American Actuarial Journal, 24(2), 168–186, 2020

ISSN: 1092-0277 print / 2325-0453 online
DOI: 10.1080/10920277.2019.1575242

Routledge
Taylor & Francis Group

Check for updates

Data Clustering with Actuarial Applications

Guojun Gan and Emiliano A. Valdez
Department of Mathematics, University of Connecticut, Storrs, Connecticut

Data clustering refers to the process of dividing a set of objects into homogeneous groups or clusters such that the objects in each cluster are more similar to each other than to those of other clusters. As one of the most popular tools for exploratory data analysis, data clustering has been applied in many scientific areas. In this article, we give a review of the basics of data clustering, such as distance measures and cluster validity, and different types of clustering algorithms. We also demonstrate the applications of data clustering in insurance by using two scalable clustering algorithms, the truncated fuzzy *c*-means (TFCM) algorithm and the hierarchical *k*-means algorithm, to select representative variable annuity contracts, which are used to build predictive models. We found that the hierarchical *k*-means algorithm is efficient and produces high-quality representative variable annuity contracts.

1. INTRODUCTION

Data clustering, also known as cluster analysis, refers to the process of dividing a set of objects into homogeneous groups or clusters such that the objects in each cluster are more similar to each other than to those of other clusters (Hartigan 1975; Jain and Dubes 1988; Kaufman and Rousseeuw 1990; Mirkin 1996; Gan, Ma, and Wu 2007; Kogan 2007; Xu and Wunsch 2008; Everitt et al. 2011; Aggarwal and Reddy 2013; Kassambara 2017). First originating in anthropology and psychology in the 1930s (Driver and Kroeber 1932; Zubin 1938; Tryon 1939), data clustering is now one of the most popular tools for exploratory data analysis and has been applied in many scientific areas, including engineering, computer science, life and medical sciences, astronomy and earth sciences, and social sciences.

Data clustering is considered a major task of data mining (Berry and Linoff 2000; Bramer 2013). Table 1 shows four major tasks of data mining. These tasks are divided into two categories based on the types of data: labeled and unlabeled. Labeled data have a specially designated attribute and the aim is to use the given data to predict the value of that attribute, for new data. Unlabeled data do not have such a designated attribute. The first two data mining tasks, association rule learning and clustering, work with unlabeled data and are known as unsupervised learning. Association rule learning concerns finding interesting relationships and correlations that exist among the values of variables (Bramer 2013; Aggarwal 2015). The last two data mining tasks, classification and numerical prediction, work with labelled data and are called supervised learning. Classification is one type of supervised learning where the designated attribute (i.e., the label) is categorical. Numerical prediction, also known as regression, is another type of supervised learning where the designated attribute is numerical (Frees 2009; Bramer 2013).

Data clustering has been applied in actuarial science. For example, Campbell (1986) applied cluster analysis to identify groups of car models with similar technical attributes for the purpose of estimating risk premium for individual car models. Yao (2016) explored territory clustering for ratemaking in motor insurance. In the discussion paper (Institute and Faculty of Actuaries 2018), Pryor mentioned using the *k*-means clustering algorithm to find earnings progression patterns in a large pension dataset. O'Hagan and Ferrari (2017) used data clustering to compress variable annuities to make nested stochastic simulations practical to run. In Gan and Valdez (2016) and Gan and Huang (2017), the authors used data clustering to select representative policies to build predictive models for valuing large portfolios of variable annuity contracts. Figure 1 shows a data mining framework proposed by Gan and Huang (2017) for valuing a large portfolio of variable annuity contracts. In this data mining framework, data clustering is used to select representative variable annuity contracts from a portfolio of contracts.

Address correspondence to Guojun Gau, Department of Mathematics, University of Connecticut, 341 Mansfield Road, Storrs, CT, 06269-1009. E-mail: emiliano.valdez@uconn.edu

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uaaj.

TABLE 1
Major Tasks of Data Mining

Unsupervised learning	Supervised learning
Data clustering	Classification
Association rules	Numerical prediction

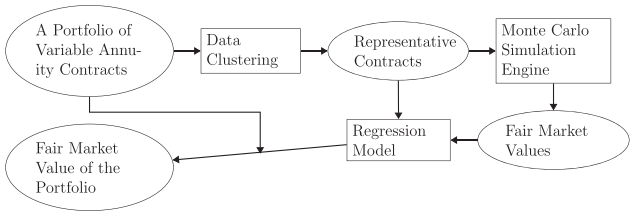

FIGURE 1. Data Mining Framework for Estimating Fair Market Values of Guarantees Embedded in Variable Annuities.

The resulting representative variable annuity contracts and their fair market values (or other quantities of interest) are used to build a predictive model, which is used to value the whole portfolio of contracts. The data mining framework has the potential to reduce the runtime of the valuation process significantly because the predictive model is much faster than the Monte Carlo valuation engine.

In this article, we give a review of data clustering and showcase its applications in actuarial science. To that end, we first describe data clustering, the notion of clusters, data types, dissimilarity measures, and cluster validity. Then we introduce several popular clustering algorithms. Finally, we illustrate the application of data clustering in the valuation of large portfolios of variable annuity contracts.

The rest of this article is organized as follows. In Section 2 we introduce data clustering in detail. In Section 3, 4 and 5 we present hierarchical, partitional, and scalable clustering algorithms. In Section 6 we apply data clustering to divide a large portfolio of variable annuity contracts into clusters and use the results to build predictive models. Section 7 concludes with some remarks.

2. DATA CLUSTERING

A typical clustering process consists of the following steps (Jain et al. 1999): pattern representation, dissimilarity measure definition, clustering, data abstraction, and output assessment. The pattern representation step involves determining the number and type of the attributes of the objects to be clustered. This step may also include feature selection and feature extraction, which refer to the process of identifying the most effective subset of the original attributes to use in clustering and the process of transforming the original attributes to new attributes, respectively. The dissimilarity measure definition step involves selecting a distance measure (distance measure and dissimilarity measure mean the same thing and are used interchangeably) that is appropriate to the data domain. The actual clustering is performed in the clustering step, where a clustering algorithm is applied to divide the data into a number of meaningful clusters. The data abstraction step involves extracting one or more prototypes from each cluster to help comprehend the clustering results. In the final step, the clustering results are assessed through some criteria.

2.1. Definition of Clusters

Although many clustering algorithms have been developed in the past several decades, there is no formal definition of clusters. In fact, it is difficult and might be misplaced to formally define clusters (Everitt et al. 2011).

There are some operational definitions of clusters. For example, Carmichael, George, and Julius (1968) suggested that a cluster is a set of data points whose distribution is continuous and relatively dense in the data space. Lorr (1983) suggested that numerical data have two kinds of clusters: compact and chained. A compact cluster consists of data points that have high mutual similarity. If any two data points in a set of data points can be connected by a path, then the set of data points forms a

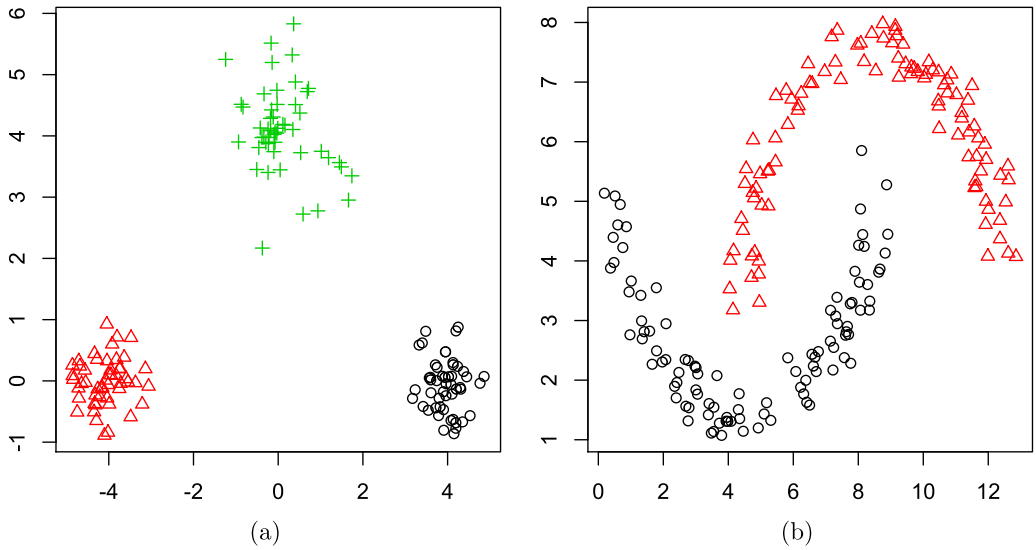

FIGURE 2. Two Datasets with Different Types of Clusters.

TABLE 2
Dataset in Tabular Form

	V_1	V_2	$\cdots$	V_d
$\mathbf{x}_1$	x_{11}	x_{12}	$\cdots$	x_{1d}
$\mathbf{x}_2$	x_{21}	x_{22}	$\cdots$	x_{2d}
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
$\mathbf{x}_n$	x_{n1}	x_{n2}	$\cdots$	x_{nd}

chained cluster. Figure 2 shows two datasets with different types of clusters. The first dataset has three compact clusters, while the second dataset has two chained clusters.

Bock (1989) also suggested the following criteria for data points in a cluster:

1. Share the same or closely related properties
2. Have small mutual distances
3. Have "contacts" or "relations" with at least one other data point in the cluster and
4. Can be clearly distinguishable from the data points that are not in the cluster.

Everitt et al. (2011) also summarized some operational definitions of clusters. One definition is that a cluster is a set of data points that are similar to each other and data points from different clusters are quite distinct.

2.2. Data Types

Data clustering algorithms typically work with standard tabular datasets that are organized as in Table 2. In the tabular data, each column represents a variable, an attribute, or a feature. Each row denotes a record, a data point, a pattern, an observation, an object, an individual, an item, or a tuple.

In general, a variable can be classified as discrete or continuous. A discrete variable usually takes on a limited number of values, while a continuous variable can take on a value between any two values. In terms of measurement scales, a variable can be categorized as nominal, ordinal, interval, and ratio. Nominal data, also called categorical data, are discrete data without a natural ordering. For example, the gender of a person is nominal. Ordinal data are discrete data with a natural order. For example, the rank of wine quality is ordinal. Interval data are continuous data with a specific order and equal intervals. An

example of interval data is temperatures. Ratio data are interval data with a natural zero. For example, the amount of money invested in a fund is ratio data.

Depending on the types of the variables, a dataset can be generally classified as discrete, continuous, or mixed-type. In a discrete dataset, all variables are discrete. In a continuous dataset, all variables are continuous. If a dataset has both discrete and continuous variables, then it is a mixed-type dataset. Clustering algorithms usually vary according to different types of datasets.

2.3. Dissimilarity Measures

Dissimilarity measures, also referred to as distance measures, play an important role in data clustering because almost all clustering algorithms rely on some distance measures to define clustering criteria. Mathematically, a distance measure D is a binary function that satisfies the following conditions (Anderberg 1973):

1. $D(\mathbf{x}, \mathbf{x}) \geq 0$ (Nonnegativity)
2. $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$ (Symmetry)
3. $D(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$ (Reflexivity)
4. $D(\mathbf{x}, \mathbf{z}) \leq D(\mathbf{x}, \mathbf{y}) + D(\mathbf{y}, \mathbf{z})$ (Triangle inequality),

where $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$ are arbitrary data points. The smaller the distance between two data points, the greater the similarity.

For continuous data, a widely used distance measure is the Minkowski distance defined by

$$D_{min}(\mathbf{x}, \mathbf{y}) = \left(\sum_{j=1}^{d} |x_j - y_j|^p \right)^{\frac{1}{p}}, \tag{1}$$

where d is the dimensionality of the dataset and $p \geq 1$. The Euclidean distance is a special case of the Minkowski distance when $p = 2$. For discrete data, a commonly used distance measure is the simple matching distance defined by

$$D_{sim}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{d} \delta(x_j, y_j), \tag{2}$$

where $\delta(\cdot, \cdot)$ is defined as

$$\delta(x_j, y_j) = \begin{cases} 0, & \text{if } x_j = y_j, \\ 1, & \text{if } x_j \neq y_j. \end{cases} \tag{3}$$

For mixed-type data, Gower (1971) proposed the following general distance measure:

$$D_{gower}(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{1}{d} \sum_{j=1}^{d} d^2(x_j, y_j)}, \tag{4}$$

where $d(x_j, y_j)$ is a distance measure for the jth variable that is defined differently for different types of variables. For ordinal and continuous attributes, $d(x_j, y_j)$ is defined as

$$d(x_j, y_j) = \frac{|x_j - y_j|}{R_j},$$

where R_j is the range of the jth attribute. For nominal attributes, $d(x_j, y_j) = \delta(x_j, y_j)$, where $\delta(\cdot, \cdot)$ is defined in Equation (3). All three measures defined above satisfy the conditions for distance measures.

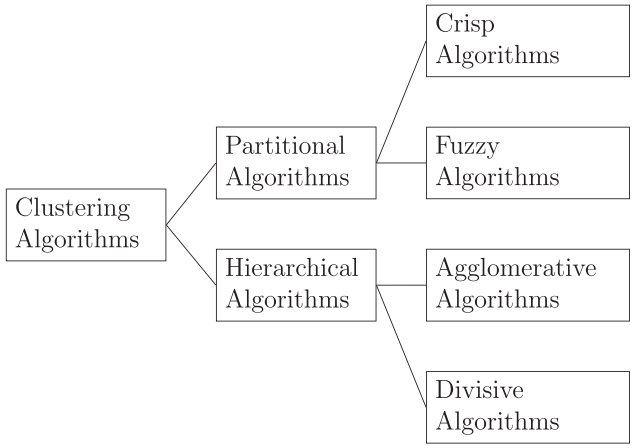

FIGURE 3. Taxonomy of Clustering Algorithms.

2.4. Taxonomy of Clustering Algorithms

Over the past several decades, many clustering algorithms have been proposed. These clustering algorithms can be divided into two categories: partitional and hierarchical clustering algorithms. A partitional clustering algorithm divides a dataset into a single partition. By contrast, a hierarchical clustering algorithm divides a dataset into a sequence of nested partitions. Figure 3 shows a diagram of different categories of clustering algorithms.

Partitional clustering algorithms can be further divided into two categories: hard and soft clustering algorithms. Hard clustering algorithms are also referred to as crisp clustering algorithms. In hard clustering, each data point belongs to exactly one cluster. Soft clustering algorithms are also referred to as fuzzy clustering algorithms. In soft clustering, a data point can belong to multiple clusters with some weights that specify the degrees of membership.

Hierarchical clustering algorithms can also be further divided into two categories: agglomerative hierarchical clustering algorithms and divisive hierarchical clustering algorithms. An agglomerative hierarchical clustering algorithm uses a bottom-up approach by starting with every data point as a cluster and repeating merging the closest pair of clusters based on some criterion until only one cluster is left. By contrast, a divisive hierarchical clustering algorithm uses a top-down approach by starting with the whole dataset as a single cluster and repeating splitting large clusters into small ones until every cluster contains only one data point.

2.5. Cluster Validity

Cluster validity refers to a collection of quantitative and qualitative measures or indices used to evaluate and assess the clustering results (Jain and Dubes 1988). There are three types of cluster validity indices (Jain and Dubes 1988; Theodoridis and Koutroubas 1999; Halkidi, Batistakis, and Vazirgiannis 2002a,b): internal, external, and relative. Internal validity indices evaluate the clustering results based only on quantities and features inherited from the underlying dataset. External validity indices evaluate the clustering results based on a prespecified structure imposed on the underlying dataset. Both internal and external validity indices are related to statistical testing. In addition, external validity indices are usually time-consuming to calculate because the Monte Carlo simulation is involved (Halkidi, Batistakis, and Vazirgiannis 2002b).

Unlike internal and external validity indices, relative validity indices evaluate the results of a clustering algorithm against the results of a different clustering algorithm or the results of the same algorithm but with different parameters. For example, the corrected Rand index (Hubert and Arabie 1985), also called the adjusted Rand index, is a popular relative validity index used to compare two partitions. The corrected Rand index between two partitions $\mathcal{C} = \{C_1, C_2, ..., C_{k_1}\}$ and $\mathcal{B} = \{B_1, B_2, ..., B_{k_2}\}$ is defined as follows (Hubert and Arabie 1985):

$$R = \frac{\binom{n}{2}\sum_{i=1}^{k_1}\sum_{j=1}^{k_2}\binom{n_{ij}}{2} - \sum_{i=1}^{k_1}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k_2}\binom{n_{\cdot j}}{2}}{\frac{1}{2}\binom{n}{2}\left[\sum_{i=1}^{k_1}\binom{n_{i\cdot}}{2} + \sum_{j=1}^{k_2}\binom{n_{\cdot j}}{2}\right] - \sum_{i=1}^{k_1}\binom{n_{i\cdot}}{2}\sum_{j=1}^{k_2}\binom{n_{\cdot j}}{2}}, \tag{5}$$

where $n_{ij} = |C_i \cap B_j|, n_{i\cdot} = |C_i|, n_{\cdot j} = |B_j|$, and n is the total number of data points. The value of R ranges from –1 to 1. If $R = 1$, then the two partitions are the same. If R is negative, then the two partitions agree by chance.

For a list of relative validity indices, readers are referred to Halkidi, Batistakis, and Vazirgiannis (2002a,b) and Gan, Ma, and Wu (2007).

3. PARTITIONAL CLUSTERING ALGORITHMS

In this section, we introduce two popular partitional clustering algorithms: the *k*-means algorithm and the fuzzy *c*-means algorithm.

3.1. *k*-Means

Among many clustering algorithms that have been developed in the past several decades, the *k*-means algorithm is perhaps the most widely used clustering algorithm due to its simplicity and efficiency. The *k*-means algorithm was independently developed by Sebestyen (1962) and Macqueen (1967) as a strategy that attempts to minimize within-group variation (Thorndike 1953; Cox 1957; Fisher 1958).

Given a set of n data points $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, the *k*-means algorithm aims to divide the dataset into k clusters by minimizing the following objective function:

$$P(U, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} u_{il} \|\mathbf{x}_i - \mathbf{z}_l\|^2, \tag{6}$$

where k is the desired number of cluster specified by the user, $U = (u_{il})_{n \times k}$ is an $n \times k$ partition matrix, $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$ is a set of cluster centers, and $\|\cdot\|$ is the L^2 norm or Euclidean distance. The partition matrix U satisfies the following conditions:

$$u_{il} \in \{0, 1\}, \quad i = 1, 2, ..., n, l = 1, 2, ..., k, \tag{7a}$$

$$\sum_{l=1}^{k} u_{il} = 1, \quad i = 1, 2, ..., n. \tag{7b}$$

The partition matrix U contains the information about the cluster memberships of the individual data points.

The *k*-means algorithm is an approximate algorithm that aims to minimize the objective function (Selim and Ismail 1984; Bobrowski and Bezdek 1991). It consists of two phases: the initialization phase and the iteration phase. In the initialization phase, k initial cluster centers are selected randomly. In the iteration phase, the algorithm repeats updating the partition matrix U and the cluster centers Z until some criterion is met. The pseudo-code of the *k*-means algorithm is shown in Algorithm 1. From the pseudo-code, we see that the *k*-means algorithm alternatively updates the partition matrix and the cluster centers in the iteration phase. Note that there are other criteria to terminate the algorithm. For example, the algorithm can be stopped if the objective function value does not change much or a maximum number of iterations is reached.

Algorithm 1: Pseudo-code of the *k*-means algorithm.

Input: A dataset X, k
Output: k clusters
1 Initialize $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k$ by randomly selecting k points from X;
2 **repeat**
3 Calculate the distance between $\mathbf{x}_i$ and $\mathbf{z}_j$ for all $1 \le i \le n$ and $1 \le j \le k$;
4 Update the partition matrix U according to Theorem 3.1;
5 Update cluster centers Z according to Theorem 3.2;
6 **until** *No further changes of the partition matrix*;
7 Return the partition matrix U and the cluster centers Z;

Theorem 3.1 *Let the cluster centers $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$ be fixed. Then the objective function given in Equation (6) is minimized if and only if*

$$u_{il} = \begin{cases} 1, & \text{if } \|\mathbf{x}_i - \mathbf{z}_l\| = \min_{1 \le j \le k} \|\mathbf{x}_i - \mathbf{z}_j\|; \\ 0, & \text{if otherwise}, \end{cases}$$

for $i = 1, 2, ..., n$ and $l = 1, 2, ..., k$.

Theorem 3.2 *Let the partition matrix U be fixed. Then the objective function given in Equation (6) is minimized if and only if*

$$z_{lj} = \frac{\sum_{i=1}^{n} u_{il} x_{ij}}{\sum_{i=1}^{n} u_{il}}, \quad l = 1, 2, ..., k, j = 1, 2, ..., d,$$

where z_{lj} is the jth component of $\mathbf{z}_l$, x_{ij} is the jth component of $\mathbf{x}_i$, and d is the dimensionality of the dataset.

3.2. Fuzzy *c*-Means

The fuzzy c-means (FCM) algorithm (Dunn 1973; Bezdek, Ehricle, and Full 1984) is a popular fuzzy clustering algorithm. The FCM algorithm has some advantages over the k-means algorithm. For example, the FCM algorithm can reduce the number of local minima of the objective function (Klawonn 2004).

The FCM algorithm is formulated to minimize the following objective function:

$$Q(U, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} u_{il}^{\alpha} \|\mathbf{x}_i - \mathbf{z}_l\|^2, \tag{8}$$

where $U = (u_{il})_{n \times k}$ is a $n \times k$ fuzzy k partition matrix, $\alpha > 1$ is the fuzzifier, $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$ is a set of k centers, and $\|\cdot\|$ is the L^2-norm or Euclidean distance. Here a fuzzy k partition of a dataset of n points is a $n \times k$ matrix that satisfies the following conditions:

$$u_{il} \in [0, 1], \quad i = 1, 2, ..., n, l = 1, 2, ..., k, \tag{9a}$$

$$\sum_{l=1}^{k} u_{il} = 1, \quad i = 1, 2, ..., n, \tag{9b}$$

$$\sum_{i=1}^{n} u_{il} > 0, \quad l = 1, 2, ..., k. \tag{9c}$$

Similar to the k-means algorithm, the FCM algorithm employs an iterative process to minimize the objective function. Algorithm 2 shows the pseudo-code of the FCM algorithm. In addition to the parameter k, the FCM algorithm requires the parameter α, which is called the fuzzifier. When $\alpha \to 1$, we will have $u_{il} \to \frac{1}{k}$. In this case, a data point belongs to all clusters with equal memberships. When $\alpha \to \infty$, one of $u_{i1}, u_{i2}, ..., u_{ik}$ will approach 1 and all others will approach 0. In this case, the FCM algorithm degenerates to the k-means algorithm.

Algorithm 2: Pseudo-code of the fuzzy c-means algorithm.

Input: A dataset X, k, α, ϵ
Output: The fuzzy partition matrix, k cluster centers
Initialize $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k$ by randomly selecting k points from X;
repeat
  Calculate the distance between $\mathbf{x}_i$ and $\mathbf{z}_j$ for all $1 \le i \le n$ and $1 \le j \le k$;
  Update the fuzzy partition matrix U according to Theorem 3.3;
  Update cluster centers Z according to Theorem 3.4;
until *The maximum change of the elements of U is less than ϵ;*
Return the fuzzy partition matrix U and the cluster centers Z;

Theorem 3.3 *Let the cluster centers $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$ be fixed. Then the objective function given in Equation (8) is minimized if and only if*

$$u_{il} = \frac{\|\mathbf{x}_i - \mathbf{z}_l\|^{-\frac{2}{\alpha-1}}}{\sum_{s=1}^{k} \|\mathbf{x}_i - \mathbf{z}_s\|^{-\frac{2}{\alpha-1}}}$$

for $i = 1, 2, ..., n$ and $l = 1, 2, ..., k$.

Theorem 3.4 *Let the partition matrix U be fixed. Then the objective function given in Equation (6) is minimized if and only if*

$$z_{lj} = \frac{\sum_{i=1}^{n} u_{il}^{\alpha} x_{ij}}{\sum_{i=1}^{n} u_{il}^{\alpha}}, \quad l = 1, 2, ..., k, j = 1, 2, ..., d,$$

where z_{lj} is the jth component of $\mathbf{z}_l$, x_{ij} is the jth component of $\mathbf{x}_i$, and d is the dimensionality of the dataset.

4. HIERARCHICAL CLUSTERING ALGORITHMS

In this section, we introduce some hierarchical clustering algorithms. Broadly speaking, this class of algorithms subdivide the dataset into a sequence of nested partitions.

4.1. Agglomerative Hierarchical Algorithms

Agglomerative hierarchical clustering algorithms are bottom-up algorithms that start with every single data point as a cluster and repeat merging clusters until only one cluster is left. Algorithm 3 shows the pseudo-code of an agglomerative hierarchical algorithm. From the pseudo-code, we see that agglomerative hierarchical clustering algorithms require a way to measure the distance between clusters to decide which two clusters to merge at each step.

Algorithm 3: Pseudo-code of an agglomerative hierarchical algorithm.

Input: A dataset X
Output: Nested partitions
Let C_i be the cluster containing only $\mathbf{x}_i$ for $i = 1, 2, ..., n$;
Calculate the distance between C_i and C_j for all $1 \leq i \leq n$ and $1 \leq j \leq n$;
repeat
    Merge two clusters that have the minimum distance to form a new cluster;
    Calculate the distances between the new cluster and the remaining clusters
until *Only one cluster is left*;
Return the nested partitions;

Lance and Williams (1967) proposed a recurrence formula that calculates the distance between a cluster and another cluster formed by the fusion of two clusters. Let C_i, C_j, and C_k be three clusters. Let $C_i \cup C_j$ be the cluster formed by merging C_i and C_j. The Lance-Williams formula calculates the distance between C_k and $C_i \cup C_j$ as

$$D(C_k, C_i \cup C_j) = \alpha_i D(C_k, C_i) + \alpha_j D(C_k, C_j) + \beta D(C_i, C_j) + \gamma |D(C_k, C_i) - D(C_k, C_j)|, \tag{10}$$

where α_i, α_j, β, and γ are parameters. According to different settings of the parameters, agglomerative hierarchical clustering algorithms can be further divided into single-linkage, complete-linkage, group average, weighted group average, centroid, median, and Ward's methods, as shown in Table 3.

The single-linkage algorithm is one of the simplest hierarchical clustering algorithm. This algorithm was first proposed by Florek et al. (1951) and then independently by McQuitty (1957) and Sneath (1957). From the Lance-Williams formula, we see that the single-linkage algorithm calculates the distance between two clusters as

TABLE 3
Commonly Used Parameters for Lance-Williams Formula, Where n_i, n_j, and n_k Denote Number of Data Points in C_i, C_j, and C_k, Respectively

Algorithm	α_i	α_j	β	γ
Single-linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete-linkage	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group average	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	0	0
Weighted group average	$\frac{1}{2}$	$\frac{1}{2}$	0	0
Centroid	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$-\frac{n_i n_j}{(n_i+n_j)^2}$	0
Median	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0
Ward's	$\frac{n_i+n_k}{n_i+n_j+n_k}$	$\frac{n_j+n_k}{n_i+n_j+n_k}$	$-\frac{n_k}{n_i+n_j+n_k}$	0

$$\begin{aligned} D(C_k, C_i \cup C_j) &= \frac{1}{2}D(C_k, C_i) + \frac{1}{2}D(C_k, C_j) - \frac{1}{2}|D(C_k, C_i) - D(C_k, C_j)| \\ &= \min\{D(C_k, C_i), D(C_k, C_j)\}. \end{aligned} \tag{11}$$

As a result, the single-linkage algorithm is also referred to as the nearest neighbor clustering algorithm, the minimum algorithm, and the connectedness algorithm (Rohlf 1982).

The complete-linkage algorithm uses the furthest neighbor distance to measure the dissimilarity of two clusters because from the Lance-Williams formula we have

$$\begin{aligned} D(C_k, C_i \cup C_j) &= \frac{1}{2}D(C_k, C_i) + \frac{1}{2}D(C_k, C_j) + \frac{1}{2}|D(C_k, C_i) - D(C_k, C_j)| \\ &= \max\{D(C_k, C_i), D(C_k, C_j)\}. \end{aligned} \tag{12}$$

The group average algorithm uses the average distances between the points in two clusters to measure the dissimilarity between the two clusters. To see this, we suppose that

$$D(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{\mathbf{x} \in C, \mathbf{y} \in C'} D(\mathbf{x}, \mathbf{y}),$$

where C and C' are two nonempty, nonoverlapping clusters and $D(\mathbf{x}, \mathbf{y})$ denotes the distance between two points. Then from the Lance-Williams formula, we have

$$\begin{aligned} D(C_k, C_i \cup C_j) &= \frac{|C_i|}{|C_i| + |C_j|} \frac{\sum_{\mathbf{x} \in C_k, \mathbf{y} \in C_i} D(\mathbf{x}, \mathbf{y})}{|C_k| \cdot |C_i|} + \frac{|C_j|}{|C_i| + |C_j|} \frac{\sum_{\mathbf{x} \in C_k, \mathbf{y} \in C_j} D(\mathbf{x}, \mathbf{y})}{|C_k| \cdot |C_j|} \\ &= \frac{1}{(|C_i| + |C_j|)|C_k|} \sum_{\mathbf{x} \in C_k, \mathbf{y} \in C_i \cup C_j} D(\mathbf{x}, \mathbf{y}). \end{aligned} \tag{13}$$

The weighted group average algorithm updates the distances in a similar way as does the group average algorithm. However, the weighted group average ignores the sizes of the clusters merged to form the new cluster. From the Lance-Williams formula, the weighted group average algorithm calculates the distance as follows:

$$D(C_k, C_i \cup C_j) = \frac{1}{2}D(C_k, C_i) + \frac{1}{2}D(C_k, C_j). \tag{14}$$

The previous four agglomerative hierarchical algorithms are called graph algorithms because the distance between a cluster and the newly formed cluster depends only on the distance between the cluster and the two clusters merged to form the new cluster. Unlike these graph algorithms, the last three agglomerative hierarchical algorithms (i.e., centroid, median, and Ward's) are called geometric algorithms.

Under the centroid algorithm, we can show that the distance between two nonempty, nonoverlapping clusters is calculated as

$$\begin{aligned} & D(C, C') \\ &= \frac{1}{|C| \cdot |C'|} \sum_{\mathbf{x} \in C} \sum_{\mathbf{y} \in C'} d(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C|^2} \sum_{\mathbf{x} \in C} \sum_{\mathbf{y} \in C} D(\mathbf{x}, \mathbf{y}) - \frac{1}{2|C'|^2} \sum_{\mathbf{x} \in C'} \sum_{\mathbf{y} \in C'} D(\mathbf{x}, \mathbf{y}) \\ &= \frac{1}{2|C|^2 |C'|^2} \sum_{\mathbf{x}_1 \in C} \sum_{\mathbf{x}_2 \in C} \sum_{\mathbf{y}_1 \in C'} \sum_{\mathbf{y}_2 \in C'} \left[D(\mathbf{x}_1, \mathbf{y}_1) + D(\mathbf{x}_2, \mathbf{y}_2) - D(\mathbf{x}_1, \mathbf{x}_2) - D(\mathbf{y}_1, \mathbf{y}_2) \right]. \end{aligned} \tag{15}$$

If squared Euclidean distance is used, the above equation becomes

$$D(C, C') = \sum_{j=1}^{d} \left(\frac{1}{|C|} \sum_{\mathbf{x} \in C} x_j - \frac{1}{|C'|} \sum_{\mathbf{y} \in C'} y_j \right)^2,$$

where d is the dimensionality of the dataset.

In the centroid algorithm, the distance between two clusters depends on the size of the clusters. If the sizes of the two clusters to be merged are quite different, then the centroid of the new cluster will be very close to that of the larger cluster. The median algorithm was proposed by Gower (1967) to alleviate the disadvantage of the centroid algorithm.

The Ward's method minimizes the loss of information associated with merging two clusters (Ward 1963; Ward and Hook 1963), where the loss of information is measured by the sum of squared errors. If the squared Euclidean distance is used, the sum of squared errors for a cluster C is calculated as

$$SSE(C) = \sum_{j=1}^{d} \sum_{\mathbf{x} \in C} (x_j - \mu_j)^2, \tag{16}$$

where μ_j is the jth component of the center of C:

$$\mu_j = \frac{1}{|C|} \sum_{\mathbf{x} \in C} x_j.$$

For examples of various agglomerative hierarchical clustering algorithms, readers are referred to Gan, Ma, and Wu (2007, Chapter 7).

4.2. Divisive Hierarchical Algorithms

Unlike agglomerative hierarchical clustering algorithms, divisive hierarchical clustering algorithms use a top-down approach to construct the nested partitions. At each step of a divisive algorithm, a cluster is split into two, and the number of clusters is increased by one.

Divisive hierarchical algorithms are usually time-consuming because there are many nontrivial ways to divide a cluster. In fact, there are $2^{|C|}-1$ nontrivial different ways to divide a cluster C. However, it is possible to construct divisive algorithms without enumerating all possible divisions. For example, DIANA (divisive analysis) (Kaufman and Rousseeuw 1990) is an example of divisive hierarchical clustering algorithms.

5. SCALABLE CLUSTERING ALGORITHMS

Most of the existing clustering algorithms focus on dividing a dataset into a small number of clusters. As the size of a dataset grows, the number of clusters into which people wish to partition also grows. In some situations, clustering a dataset is the

preliminary step of data analysis. In these cases, the clustering results are used as input in subsequent steps to build predictive models, which require a large number of clusters to produce accurate predictions (see, e.g., Gan and Huang 2017). In this section, we introduce some clustering algorithms that are efficient in dividing a large dataset into a large number of clusters.

5.1. TFCM

The TFCM (truncated fuzzy c-means) is an algorithm proposed by Gan, Lau, and Ma (2016) as an extension of the FCM algorithm to divide a large dataset into a large number of clusters in an efficient way. The main idea behind the TFCM algorithm is to reduce the number of distance calculations during the iterative process of the FCM algorithm.

To describe the TFCM algorithm, we let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ be a dataset containing n data points. Let k be the desired number of clusters. Let T be an integer such that $1 \leq T \leq k$ and let $\mathcal{U}_T$ be the set of fuzzy partition matrices U such that each row of U has at most T nonzero entries, that is, $U \in \mathcal{U}_T$ if U satisfies the following conditions given in Equation (9) and

$$|\{l : u_{il} > 0\}| \leq T, \quad i = 1, 2, ..., n, \tag{17}$$

where $|\cdot|$ denotes the number of elements in a set.

The TFCM algorithm and the FCM algorithm share the same objective function. However, the constraints of the objective function are different as mentioned above. The goal of the TFCM algorithm is to find a truncated fuzzy partition matrix U and a set of cluster centers Z that minimizes the following objective function:

$$P(U, Z) = \sum_{i=1}^{n} \sum_{l=1}^{k} u_{il}^{\alpha} \left(\|\mathbf{x}_i - \mathbf{z}_l\|^2 + \epsilon \right), \tag{18}$$

where $\alpha > 1$ is the fuzzifier, $U \in \mathcal{U}_T, Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$ is a set of cluster centers, $\|\cdot\|$ is the L^2-norm or Euclidean distance, and ϵ is a small positive number used to prevent division by zero.

To solve the optimization problem, the TFCM algorithm uses an alternative updating scheme shown in Algorithm 2. Since the constraints of the FCM algorithm and the TFCM algorithm are different, how the fuzzy partition matrix and the cluster centers are updated is also different. For the TFCM algorithm, Theorem 5.1 and Theorem 5.2 describe how to update the fuzzy membership U and how to update the cluster centers Z, respectively.

Theorem 5.1 *Let the cluster centers Z be fixed. Then the fuzzy partition matrix $U \in \mathcal{U}_T$ that minimizes the objective function (18) is calculated by*

$$u_{il} = \frac{\left(\|\mathbf{x}_i - \mathbf{z}_l\|^2 + \epsilon \right)^{-\frac{1}{\alpha - 1}}}{\sum_{s \in I_i} \left(\|\mathbf{x}_i - \mathbf{z}_s\|^2 + \epsilon \right)^{-\frac{1}{\alpha - 1}}}, \quad 1 \leq i \leq n, l \in I_i, \tag{19}$$

where I_i is the set of indices of the T centers that are closest to $\mathbf{x}_i$.

Theorem 5.2 *Let the fuzzy partition matrix $U \in \mathcal{U}_T$ be fixed. Then the set of centers Z that minimizes the objective function (18) is calculated by*

$$z_{lj} = \frac{\sum_{i=1}^{n} u_{il}^{\alpha} x_{ij}}{\sum_{i=1}^{n} u_{il}^{\alpha}} = \frac{\sum_{i \in C_l} u_{il}^{\alpha} x_{ij}}{\sum_{i \in C_l} u_{il}^{\alpha}}, \tag{20}$$

for $l = 1, 2, ..., k$ and $j = 1, 2, ..., d$, where d is the dimension of the dataset, z_{lj} is the jth component of $\mathbf{z}_l$, and $C_l = \{i : u_{il} > 0\}$.

5.2. Hierarchical *k*-Means

The traditional k-means algorithm is extremely slow when used to divide a large dataset into a large number of clusters due to the large number of distance calculations in each iteration. To address this scalability issue, hierarchical k-means (Nister and Stewenius 2006) uses a divisive approach to apply the traditional k-means with small k's repeatedly until the desired number of clusters is reached.

Algorithm 4: Pseudo-code of the hierarchical k-means algorithm.

Input: A dataset X, k
Output: k clusters
Apply the k-means algorithm to divide the dataset into two clusters;
repeat
    Apply the k-means algorithm to divide the largest existing cluster into two clusters;
until *The number of clusters is equal to* k;
Return the k clusters;

Algorithm 4 shows the pseudo-code of the hierarchical k-means algorithm. In this hierarchical k-means algorithm, we divide an existing cluster into two at each step. The clustering result is similar to a binary tree.

6. APPLICATION IN VARIABLE ANNUITY VALUATION

In this section, we illustrate the use of data clustering in the valuation of large portfolios of variable annuities (VAs). Data clustering can be effectively used to select representative contracts for metamodeling, which has been demonstrated to be useful for the valuation of large VA portfolios. Here we focus on using scalable clustering algorithms to select representative VA contracts. We will evaluate the quality of the representative contracts using some validity measures as well as the prediction accuracy by a predictive model.

6.1. Description of the Problem

A variable annuity is a tax-deferred retirement vehicle created by insurance companies to address concerns that many people have about outliving their assets. Under a VA contract, the policyholder agrees to make one lump-sum or a series of purchase payments to the insurer and the insurer agrees to make benefit payments to the policyholder, beginning either immediately or at a future date. The policyholder invests the premiums in a number of mutual funds provided by the insurer.

One major feature of VAs is that they contain guarantees. Common guarantees include guaranteed minimum death benefit (GMDB), guaranteed minimum withdrawal benefit (GMWB), guaranteed minimum income benefit (GMIB), and guaranteed minimum maturity benefit (GMMB). Because of these attractive guarantee features, lots of variable annuity contracts have been sold in the past two decades. However, these are financial guarantees that cannot be adequately addressed by traditional actuarial approaches (Hardy 2003).

Many insurance companies adopted dynamic hedging to mitigate the financial risks associated with the guarantees embedded in VAs. Dynamic hedging requires calculating the fair market values and Greeks (i.e., sensitivities of the fair market values of the guarantees to major market indices) for every VA contract. Since the guarantees are complex, there is no closed-form formula to calculate the fair market values and the Greeks. In practice, insurance companies resort to Monte Carlo simulation to calculate the fair market values and Greeks of these guarantees.

However, Monte Carlo simulation is computationally intensive for valuing the guarantees for a large portfolio of VAs (Dardis 2016). In fact, using Monte Carlo simulation to calculate the fair market values of a large portfolio of VAs may take hours or several days. Recently the metamodeling approach has been proposed to address the computational problem. See, for example, Hejazi and Jackson (2016), Gan and Huang (2017), Gan and Valdez (2017a), Hejazi, Jackson, and Gan (2017), Gan (2018), Gan and Valdez (2018), and Xu et al. (2018). The metamodeling approach consists of the following major steps: (1) selecting a small number of representative contracts, (2) using Monte Carlo simulation to calculate the fair market values (or other quantities of interest) of the representative contracts, (3) building a regression model (i.e., the metamodel) based on the representative contracts and their fair market values, and (4) finally using the regression model to value the whole portfolio of variable annuity contracts. The main idea of metamodeling techniques is to construct a regression model based on a small number of representative VA contracts to reduce the number of contracts that are valued by Monte Carlo simulation.

6.2. Description of the Data

To demonstrate the application of data clustering in VA valuation, we use a synthetic dataset created in Gan and Valdez (2017b). The dataset contains 190,000 synthetic VA policies, each of which is described by 45 variables. Some of the variables have identical values and thus are not useful for building predictive models. We exclude these variables from the clustering

step as well as from the predictive model. The explanatory variables used to select representative VA contracts and build the predictive model include the following:

- gender: Gender of the policyholder
- productType: Product type of the VA policy
- gmwbBalance: GMWB balance
- gbAmt: Guaranteed benefit amount,
- withdrawal: Total amount withdrawn
- FundValue$i$: Account value of the ith fund, for $i = 1, 2, \ldots, 10$
- age: Age of the policyholder and
- ttm: Time to maturity in years.

Table 4 shows the summary statistics of the explanatory variables as well as the fair market values. Table 4(a) shows the summary statistics of the continuous explanatory variables. Policyholders can select from 10 different investment funds. From the summary statistics, we see that there are many zeros for the investment funds. The reason is that many policyholders do not invest in all available funds. Table 4(b) shows the counts of the categorical variables. We see the distribution of female and male; there are about 40% female and 60% male for each product type. However, the number of policies in each product type is the same, that is, 10,000 for each of the 19 product types.

The fair market values of the guarantees are calculated by a simple Monte Carlo simulation model (Gan 2015; Gan and Valdez 2017b). The summary statistics are shown in Table 4(c), and a histogram of the fair market values is shown in Figure 4. From the table, we see that there are negative fair market values. Since the fair market value for a VA contract is equal to the present value of benefits minus that of the fees, it is negative when the present value of benefits is less than that of the fees. From Figure 4, we see that the distribution of the fair market values is positively skewed.

6.3. Clustering Results

We apply the TFCM algorithm and the hierarchical k-means to divide the portfolio of VA contracts into 340 and 680 clusters. Following the previous studies in Gan and Lin (2017), we set the initial number of clusters to be 10 times the number of explanatory variables (including dummy binary variables) used to build predictive models. Then we test the clustering algorithms again by doubling the initial number of clusters.

Since the portfolio of VA contracts does not contain any cluster labels (i.e., we do not know the clusters to which the VA contracts belong), we use the relative within cluster sum of squares (RWCSS) to assess the accuracy of a single clustering result. The RWCSS measure is defined as

$$RWCSS = \frac{\sum_{l=1}^{k} \sum_{\mathbf{x} \in C_l} \sum_{j=1}^{d} (x_j - z_{lj})^2}{\sum_{\mathbf{x} \in X} \sum_{j=1}^{d} (x_j - \bar{x}_j)^2}, \tag{21}$$

where C_l denotes the lth cluster, $\mathbf{z}_l$ is the center of the lth cluster, and $\bar{\mathbf{x}}$ is the center of the whole dataset X. The RWCSS is the ratio of the within cluster sum of squares when the whole dataset is divided into k clusters over that when the whole dataset is in a single cluster. When $k = 1$, we have $RWCSS = 1$. When k is equal to the number of data points and each data point forms a cluster, we have $RWCSS = 0$. Given the same number of clusters, the lower the $RWCSS$, the better the clustering result.

Table 5 shows the $RWCSS$ measures and the runtime of the two clustering algorithms with different values for k. From the table, we see that the TFCM algorithm outperforms the hierarchical k-means given the same k. For example, the $RWCSS$ obtained by TFCM with $k = 340$ is 0.82, which is lower than 0.90, which is the $TWCSS$ obtained by the hierarchical k-means with $k = 340$. In terms of speed, the hierarchical k-means is faster than the TFCM algorithm by an order of magnitude. The reason is that the TFCM algorithm spends much of time in the initialization phase and in the iterative phase for sorting.

To compare the four clustering results obtained by the two clustering algorithms with different number of clusters (k), we calculate the corrected Rand indices between all pairs of the clustering results. The corrected Rand indices are shown in Table 6. From the table, we see that the clustering results are similar as the indices are larger than 0.5.

TABLE 4
Summary Statistics of Explanatory Variables and Fair Market Values
(a) Summary Statistics of Continuous Variables

	Min	1st Q.	Median	3rd Q.	Max
gmwbBalance	0.00	0.00	0.00	0.00	499,708.73
gbAmt	50,001.72	179,451.09	303,003.73	426,821.47	989,204.53
withdrawal	0.00	0.00	0.00	0.00	499,585.73
FundValue1	0.00	0.00	8,147.13	38,646.82	916,827.66
FundValue2	0.00	0.00	8,242.02	37,914.72	844,322.70
FundValue3	0.00	0.00	4,833.63	23,886.98	580,753.42
FundValue4	0.00	0.00	4,140.30	20,435.29	483,936.90
FundValue5	0.00	0.00	7,108.14	31,635.25	494,381.61
FundValue6	0.00	0.00	8,378.14	38,679.09	861,030.03
FundValue7	0.00	0.00	6,468.98	30,629.72	629,146.30
FundValue8	0.00	0.00	6,127.67	28,975.36	553,867.27
FundValue9	0.00	0.00	5,826.50	27,674.85	659,807.39
FundValue10	0.00	0.00	6,617.43	30,791.50	588,961.66
age	34.52	42.03	49.45	56.96	64.46
ttm	0.59	10.34	14.51	18.76	28.52

(b) Counts of Categorical Variables

	gender			gender	
productType	F	M	productType	F	M
ABRP	4068	5932	IBRP	4007	5993
ABRU	3974	6026	IBRU	4027	5973
ABSU	4054	5946	IBSU	4007	5993
DBAB	3974	6026	MBRP	3909	6091
DBIB	3948	6052	MBRU	3992	6008
DBMB	4013	5987	MBSU	3980	6020
DBRP	4002	5998	WBRP	3970	6030
DBRU	3952	6048	WBRU	4076	5924
DBSU	4038	5962	WBSU	3994	6006
DBWB	4022	5978			

(c) Summary Statistics of Fair Market Values

	Min	1st Q.	Median	3rd Q.	Max
fmv	(46,973.43)	29,388.71	81,814.27	167,665.95	2,517,911.77

6.4. Predictive Modeling Results

We also evaluate the quality of the clustering results produced by the two clustering algorithms based on the predictive models that use the clustering results as inputs. In particular, we use the ordinary kriging model to evaluate the quality of the clustering results. The ordinary kriging model has been used to predict fair market values in previous studies. See, for example, Gan and Huang (2017).

To measure the accuracy of the ordinary kriging model, we use the following two measures: the percentage error and the R^2. The percentage error measures the aggregate accuracy of the result at the portfolio level because the errors at the individual

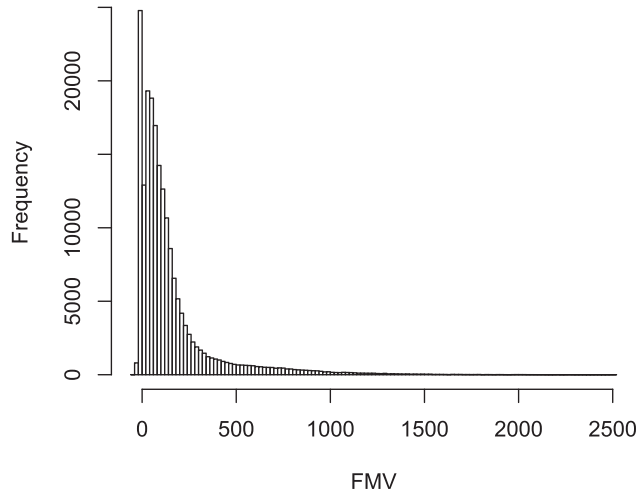

FIGURE 4. Histogram of Fair Market Values, in 1000s.

TABLE 5
Performance of TFCM Algorithm and Hierarchical *k*-Means on VA Data

	Hkmean (340)	Hkmean (680)	TFCM (340)	TFCM (680)
RWCSS	0.90	0.76	0.82	0.66
Runtime(sec)	130.02	136.19	2,647.11	5,544.81

TABLE 6
Corrected Rand Indices of Clustering Results

	Hkmean (340)	Hkmean (680)	TFCM (340)	TFCM (680)
Hkmean (340)	1.00	0.67	0.62	0.55
Hkmean (680)	0.67	1.00	0.51	0.62
TFCM (340)	0.62	0.51	1.00	0.56
TFCM (680)	0.55	0.62	0.56	1.00

contract level can offset each other. In general, the lower the absolute value of *PE*, the better the result. The R^2 measures the accuracy of the result at the individual contract level. The higher the R^2, the more accurate the result.

To define these measures, we let y_i and $\hat{y}_i$ denote the fair market value of the ith variable annuity contract obtained from the Monte Carlo simulation model and that estimated by the ordinary kriging method, respectively, for $i = 1, 2, ..., n$, where n is the total number of VA contracts in the portfolio. For the portfolio used in this article, $n = 190{,}000$. The percentage error at the portfolio level is defined as

$$PE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)}{\sum_{i=1}^{n} y_i}. \tag{22}$$

R^2 is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \mu)^2}, \tag{23}$$

where μ is the average fair market value:

TABLE 7
Accuracy and runtime of ordinary Kriging Model Based on Different Clustering Results, Numbers in Parentheses the Numbers of Clusters

	Hkmean (340)	Hkmean (680)	TFCM (340)	TFCM (680)
PE	0.02	−0.02	−0.01	−0.02
R^2	0.82	0.92	0.81	0.92
Runtime(sec)	329.50	787.11	334.62	808.99

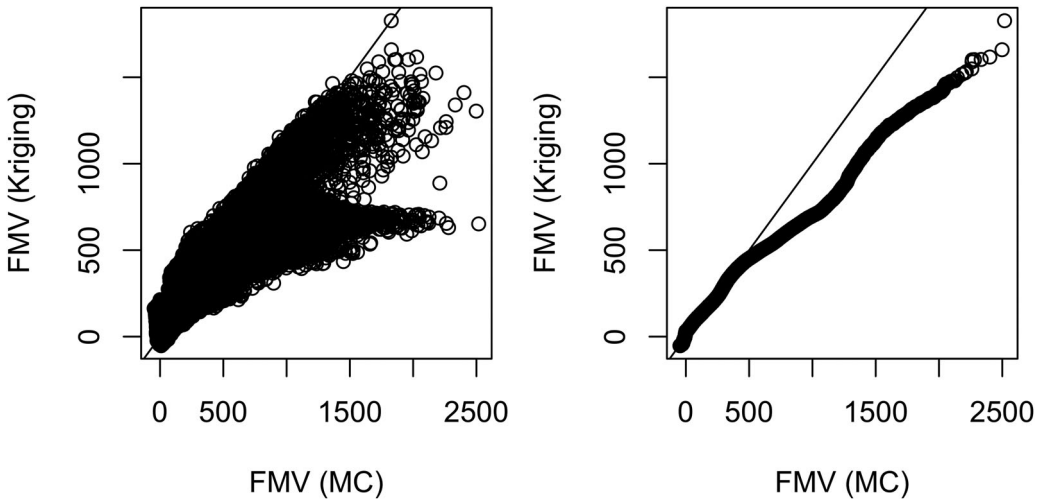

FIGURE 5. Scatter and QQ Plots of Ordinary Kriging Model Based on Clustering Result from Hierarchical k-Means with $k = 340$.

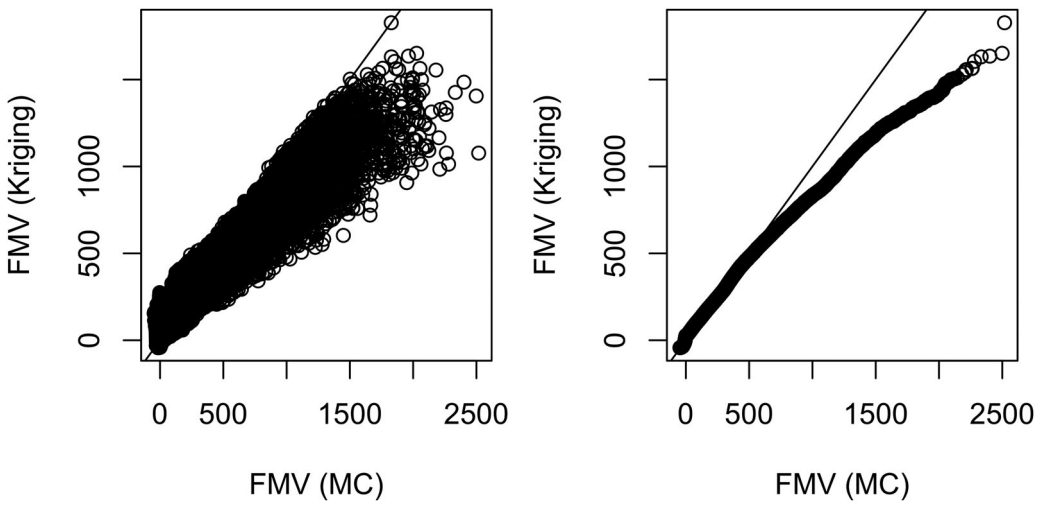

FIGURE 6. Scatter and QQ Plots of Ordinary Kriging Model Based on Clustering Result from Hierarchical k-Means with $k = 680$.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

Table 7 shows the performance of the ordinary kriging model based on different clustering results. From the table, we see that given the same number of clusters, the accuracy of the ordinary kriging model based on the clustering result of TFCM is

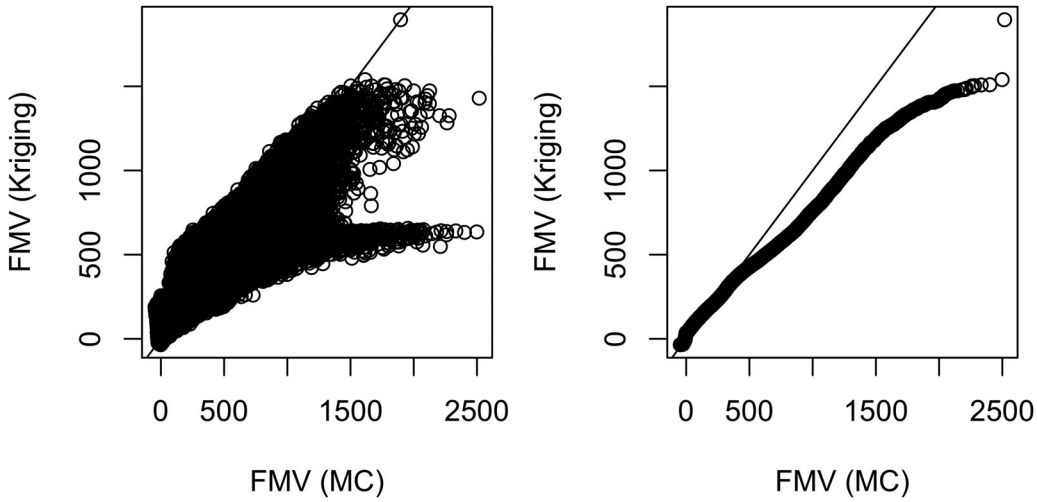

FIGURE 7. Scatter and QQ Plots of Ordinary Kriging Model Based on Clustering Result from TFCM with $k = 340$.

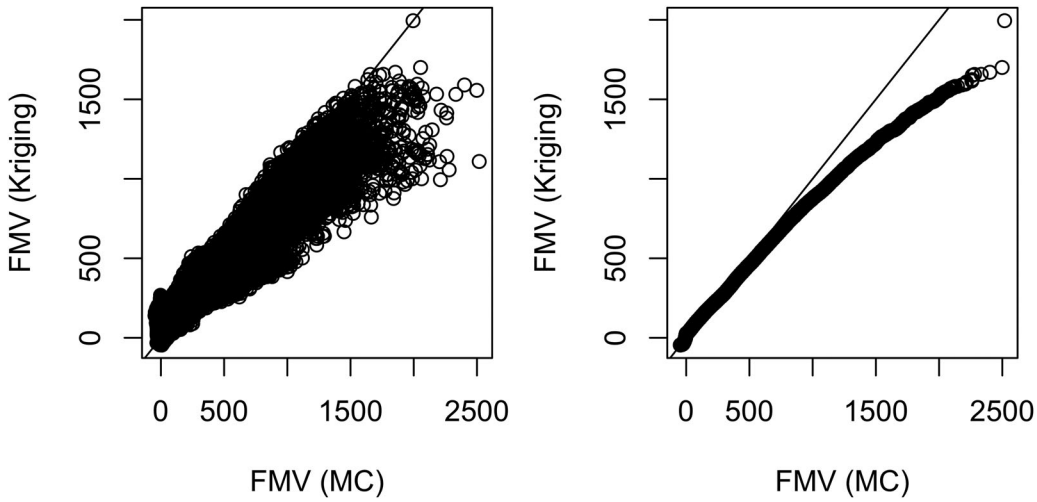

FIGURE 8. Scatter and QQ Plots of Ordinary Kriging Model Based on Clustering Result from TFCM with $k = 680$.

similar to that based on the clustering result of hierarchical k-means. For example, when $k = 340$, the R^2 based on TFCM is the same as that based on hierarchical k-means. The runtime of ordinary kriging doubles when the number of clusters doubles due to the increasing number of distance calculation.

Figures 5 and 6 show the scatter plot and the QQ plot between the fair market values calculated by Monte Carlo simulation and those estimated by ordinary kriging based on the hierarchical k-means algorithm. From the figures, we see that ordinary kriging does not fit the tail well. However, increasing the number of representative contracts leads to more accurate results. Figures 7 and 8 show the scatter plot and the QQ plot between the fair market values calculated by Monte Carlo simulation and those estimated by ordinary kriging based on the TFCM algorithm. We see similar patterns as before.

In summary, our numerical experiments show that the hierarchical k-means algorithm is superior to the TFCM algorithm when used to select representative VA contracts. The hierarchical k-means is faster than the TFCM algorithm by an order of magnitude. In addition, the quality of the representative VA contracts produced by hierarchical k-means is comparable to that of the representative VA contracts produced by the TFCM algorithm.

7. SUMMARY AND CONCLUSIONS

Data clustering is one of the most popular tools for exploratory data analysis and a major task of data mining. Since data clustering works with unlabeled data, it is also known as unsupervised learning. As a result, data clustering has been applied to

many scientific areas, including engineering, computer science, life science, and social science. In this article, we provided a review of the fundamental concepts of data clustering and some clustering algorithms. In particular, we introduced distance measures, cluster validity, and different types of clustering algorithms.

To demonstrate the applications of data clustering in actuarial science and insurance, we adopted an example that demonstrates its usefulness in the valuation of large portfolios of VA contracts. We applied two scalable clustering algorithms, the TFCM algorithm and the hierarchical *k*-means algorithm, to divide a large portfolio of variable annuity contracts into a large number of clusters, which are used to select representative contracts for building predictive models. We also evaluated the quality of the clustering results produced by the two clustering algorithms by the accuracy of the resulting predictive model. Our numerical results show that, overall, the hierarchical *k*-means outperforms the TFCM algorithm.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their helpful and constructive comments that greatly improved the article.

FUNDING

We would like to acknowledge the financial support provided by the Centers of Actuarial Excellence (CAE) grant on data mining[1] from the Society of Actuaries.

ORCID

Guojun Gan http://orcid.org/0000-0003-3285-7116

REFERENCES

Aggarwal, C. C. 2015. *Data mining: The textbook*. New York: Springer.

Aggarwal, C. C., and C. K., Reddy, editors. 2013. *Data clustering: Algorithms and applications*. Boca Raton, FL: CRC Press.

Anderberg, M. 1973. *Cluster analysis for applications*. New York: Academic Press.

Berry, M., and G. Linoff. 2000. *Mastering data mining*. New York: John Wiley & Sons.

Bezdek, J. C., R., Ehrlich, and W. Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences* 10(2–3): 191–203.

Bobrowski, L., and J. Bezdek. 1991. *c*-means clustering with the l_1 and l_∞ norms. *IEEE Transactions on Systems, Man and Cybernetics* 21(3): 545–554.

Bock, H. 1989. Probabilistic aspects in cluster analysis. In *Conceptual and numerical analysis of data*, 2nd ed. O. Opitz, 12–44, Augsburg: Springer-Verlag.

Bramer, M. 2013. *Principles of Data Mining*. 2nd ed. New York: Springer.

Campbell, M. 1986. An integrated system for estimating the risk premium of individual car models in motor insurance. *ASTIN Bulletin* 16(2): 165–183.

Carmichael, J., J., George, and R. Julius. 1968. Finding natural clusters. *Systematic Zoology* 17(2): 144–150.

Cox, D. R. 1957. Note on grouping. *Journal of the American Statistical Association*, 52(280): 543–547.

Dardis, T. 2016. Model efficiency in the U.S. life insurance industry. *The Modeling Platform* (3): 9–16.

Driver, H. E., and A. L. Kroeber. 1932. Quantitative expression of cultural relationships. *University of California Publications in American Archaeology and Ethnology* 31(4): 211–256.

Dunn, J. C. 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics* 3(3): 32–57.

Everitt, B. S., S., Landau, M., Leese, and D. Stahl. 2011. *Cluster analysis*. Hoboken, NJ: Wiley.

Fisher, W. D. 1958. On grouping for maximum homogeneity. *Journal of the American Statistical Association* 53(284): 789–798.

Florek, K., J., Lukaszewicz, H., Steinhaus, and S. Zubrzycki. 1951. Sur la liaison et la division des points d'un ensemble fini. *Colloquium Mathematicum* 2: 282–285.

Frees, E. W. 2009. *Regression modeling with actuarial and financial applications*. Cambridge: Cambridge University Press.

Gan, G. 2015. A multi-asset Monte Carlo simulation model for the valuation of variable annuities. In *Proceedings of the Winter Simulation Conference*, 3162–3163. Piscataway, NJ: IEEE Press.

Gan, G. 2018. Valuation of large variable annuity portfolios using linear models with interactions. *Risks* 6(3): 71.

Gan, G., and J. Huang. 2017. A data mining framework for valuing large portfolios of variable annuities. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1467–1475.

Gan, G., Q., Lan, and C. Ma. 2016. Scalable clustering by truncated fuzzy *c*-means. *Big Data and Information Analytics* 1(2/3): 247–259.

Gan, G., and X. S. Lin. 2017. Efficient Greek calculation of variable annuity portfolios for dynamic hedging: A two-level metamodeling approach. *North American Actuarial Journal* 21(2): 161–177.

Gan, G., C., Ma, and J. Wu. 2007. *Data clustering: Theory, algorithms, and applications*. Philadelphia: SIAM Press.

[1] https://actscidm.math.uconn.edu/

Gan, G., and E. A. Valdez. 2016. An empirical comparison of some experimental designs for the valuation of large variable annuity portfolios. *Dependence Modeling* 4(1): 382–400.
Gan, G., and E. A. Valdez. 2017a. Modeling partial greeks of variable annuities with dependence. *Insurance: Mathematics and* Econocmics 76: 118–134.
Gan, G., and E. A. Valdez. 2017b. Valuation of large variable annuity portfolios: Monte carlo simulation and synthetic datasets. *Dependence Modeling* 5: 354–374.
Gan, G., and E. A. Valdez. 2018. Regression modeling for the valuation of large variable annuity portfolios. *North American Actuarial Journal* 22(1): 40–54.
Gower, J. 1967. A comparison of some methods of cluster analysis. *Biometrics* 23(4): 623–637.
Gower, J. 1971. A general coefficient of similarity and some of its properties. *Biometrics* 27(4): 857–874.
Halkidi, M., Y., Batistakis, and M. Vazirgiannis. 2002a. Cluster validity methods: Part I. *ACM SIGMOD Record* 31(2): 40–45.
Halkidi, M., Y., Batistakis, and M. Vazirgiannis. 2002b. Clustering validity checking methods: Part II. *ACM SIGMOD Record* 31(3): 19–27.
Hardy, M. 2003. *Investment guarantees: Modeling and risk management for equity-linked life insurance*. Hoboken, NJ: John Wiley and Sons.
Hartigan, J. A. 1975. *Clustering algorithms*. Probability & Mathematical Statistics. New York: Wiley.
Hejazi, S. A., and K. R. Jackson. 2016. A neural network approach to efficient valuation of large portfolios of variable annuities. *Insurance: Mathematics and Economics* 70: 169–181.
Hejazi, S. A., K. R., Jackson, and G. Gan. 2017. A spatial interpolation framework for efficient valuation of large portfolios of variable annuities. *Quantitative Finance and Economics* 1(2): 125–144.
Hubert, L., and P. Arabie. 1985. Comparing partitions. *Journal of Classification* 2: 193–218.
Institute and Faculty of Actuaries. 2018. What data science means for the future of the actuarial profession: Abstract of the london discussion. *British Actuarial Journal* 23: e16.
Jain, A., M., Murty, and P. Flynn. 1999. Data clustering: A review. *ACM Computing Surveys* 31(3): 264–323.
Jain, A. K., and R. C. Dubes. 1988. *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall.
Kassambara, A. 2017. *Practical guide to cluster analysis in R: Unsupervised machine learning*. CreateSpace Independent Publishing Platform.
Kaufman, L., and P. J. Rousseeuw. 1990. *Finding groups in data: An introduction to cluster analysis*. Hoboken, NJ: Wiley.
Klawonn, F. 2004. Fuzzy clustering: Insights and a new approach. *Mathware & Soft Computing* 11: 125–142.
Kogan, J. 2007. *Introduction to clustering large and high-dimensional data*. Cambridge: Cambridge University Press.
Lance, G., and W. Williams. 1967. A general theory of classificatory sorting strategies I. Hierarchical systems. *Computer Journal* 9(4): 373–380.
Lorr, M. 1983. *Cluster analysis for social scientists*. Jossey-Bass Social and Behavioral Science Series. San Francisco: Jossey-Bass.
Macqueen, J. 1967. Some methods for classification and analysis of multivariate observations. In, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics andProbability*, ed. L.LeCamand J.Neyman, vol. 1, 281–297, Berkely: University of California Press.
McQuitty, L. 1957. Elementary linkage analysis for isolating orthogonal and oblique types andtypal relevancies. *Educational and Psychological Measurement* 17: 207–222.
Mirkin, B. 1996. *Mathematical classification and clustering*. New York: Springer.
Nister, D., and H. Stewenius. 2006. Scalable recognition with a vocabulary tree. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2161–2168.
O'Hagan, A., and C. Ferrari. 2017. Model-based and nonparametric approaches to clustering for data compression in actuarial applications. *North American Actuarial Journal* 21(1): 107–146.
Rohlf, F. 1982. Single link clustering algorithms. In *Handbook of statistics*, ed. P. Krishnaiah and L. Kanal, vol. 2, 267–284, Amsterdam: North-Holland.
Sebestyen, G. S. 1962. Pattern recognition by an adaptive process of sample set construction. *IRE Transactions on Information Theory* 8(5): 82–91.
Selim, S., and M. Ismail. 1984. k-means-type algorithms: A generalized convergence theorem and characterizationof local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6(1): 81–87.
Sneath, P. 1957. The applications of computers to taxonomy. *Journal of General Microbiology* 17: 201–226.
Theodoridis, S., and K. Koutroubas. 1999. *Pattern recognition*. London: Academic Press.
Thorndike, R. L. 1953. Who belongs in the family? *Psychometrika* 18(4): 267–276.
Tryon, R. C. 1939. *Cluster analysis: Correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*. Ann Arbor, MI: Edwards Brothers.
Ward, J., Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association* 58(301): 236–244.
Ward, J., Jr., and M. Hook. 1963. Application of an hierarchical grouping procedure to a problem of groupingprofiles. *Educational and Psychological Measurement* 23(1): 69–81.
Xu, R., and D. Wunsch. 2008. *Clustering*. Hoboken, NJ: Wiley-IEEE Press.
Xu, W., Y., Chen, C., Coleman, and T. F. Coleman. 2018. Moment matching machine learning methods for risk management of large variable annuity portfolios. *Journal of Economic Dynamics and Control* 87: 1–20.
Yao, J. 2016. *Clustering in general insurance pricing. International Series on Actuarial Science* vol. 2159–179. Cambridge: Cambridge University Press.
Zubin, J. 1938. A technique for measuring like-mindedness. *Journal of Abnormal and Social Psychology* 33(4): 508–516.

Discussions on this article can be submitted until January 1, 2021. The authors reserve the right to reply to any discussion. Please see the Instructions for Authors found online at http://www.tandfonline.com/uaaj for submission instructions.