Applied Soft Computing Journal 96 (2020) 106639

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Metaheuristic-based possibilistic multivariate fuzzy weighted *c*-means algorithms for market segmentation

R.J. Kuo [a], Patipharn Amornnikun [a], Thi Phuong Quyen Nguyen [b,*]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei, 106, Taiwan

[b] Faculty of Project Management, The University of Danang, University of Science and Technology, 54 Nguyen Luong Bang, Danang, Viet Nam

ARTICLE INFO

Article history:
Received 26 November 2019
Received in revised form 8 August 2020
Accepted 12 August 2020
Available online 17 August 2020

Keywords:
Possibilistic multivariate fuzzy weighted *c*-means algorithm
Mixed data
Market segmentation
Metaheuristics
Sine cosine algorithm

ABSTRACT

This study proposed the metaheuristic-based possibilistic multivariate fuzzy weighted *c*-means (PMFWCM) algorithm for clustering mixed data PMFWCM algorithm itself is normally used for numerical data. To implement in the real application of market segmentation, where the data usually contains both numerical and categorical attributes model improvement is a need. First, the distance between two mixed-attribute objects is calculated by using the object–cluster similarity measure. Then, three metaheuristics, i.e., genetic algorithm (GA), particle swarm optimization algorithm (PSO), and sine cosine algorithm (SCA), are employed to integrate with the PMFWCM algorithm for cluster analysis. This combination aims to improve the clustering performance of the PMFWCM algorithm and to make the clustering results more stable. To cluster a real-world dataset certainly, the experiment with benchmark datasets from UCI machine learning repository is conducted to verify the performance of the proposed algorithms. The experiment results show that the clustering performance of the SCA-PMFWCM, GA-PMFWCM and PSO-PMFWCM algorithms are better than that of the PMFWCM algorithm. Moreover, from case study results, the SCA-PMFWCM algorithm gives the smallest sum of squared error and computational time compared with the GA-PMFWCM, PSO-PMFWCM, and PMFWCM algorithms.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Commonly, data clustering is the task of partitioning a given dataset into several distinct clusters in which the data instances within a cluster are more similar to each other and distinguish to those in other clusters [1]. Nowadays, clustering has become a popular and crucial technique since its applications are implemented in many fields such as pattern recognition [2,3], image processing [4], information retrieval [5,6], text mining [7], computer graphics [8], and so on. Regarding market segmentation, there are numerous publications that used clustering techniques to identify the market segments such as categorical data clustering in tourism market [9], fuzzy clustering in housing buyer market [10], hierarchical clustering in the emerging market for electric vehicles [11], *k*-mode clustering for e-commerce business [12], etc. In general, selecting different clustering techniques in market segmentation is based on the features of data in each market field.

Normally, data is usually divided into two different types of information: numerical or categorical. Numerical data contains measurable attributes. Moreover, the numerical data can be added together or has an ascending (least to greatest) and descending (greatest to least) order. In contrast, categorical data is grouped into some sort of category or multiple categories that cannot use the standard distance measure such as Euclidean distance to calculate the cluster center or the distance between two instances. Categorical data contains two main classes: nominal and ordinal data. Nominal categorical data describes specified information that there is no intrinsic ordering to the categories such as career (teacher, doctor or chief), or color (blue, green, or black). Ordinal categorical data has some senses of order among its values, e.g. clothes sizes (small, medium, large), education level (bachelor, master, Ph.D.). In the case that the dataset contains both numerical and categorical attributes, it can be called as mixed data [13].

The clustering algorithms for different types of data can be distinguished from distance functions and ways to find out the cluster centers. The Euclidean metric is the most popular one for numerical data. Some popular clustering algorithms use the Euclidean metric to calculate the distance between data instances and cluster centers such as *k*-means algorithm [14] and fuzzy *c*-means (FCM) algorithm [15]. Concerning the categorical data, the Hamming distance is the simplest one. Regarding mixed data,

* Corresponding author.
E-mail addresses: rjkuo@mail.ntust.edu.tw (R.J. Kuo), quyen.ntp@gmail.com (T.P.Q. Nguyen).

https://doi.org/10.1016/j.asoc.2020.106639
1568-4946/© 2020 Elsevier B.V. All rights reserved.

people try to combine the distance metric for numerical and categorical attributes with some weights for adjustment. The fuzzy *k*-prototype algorithm [16] is the most representative algorithm for mixed data.

However, real-world cases always fall into mixed attribute datasets. Therefore, this study will focus on developing clustering algorithms for mixed numerical and categorical data. The FCM algorithm is known as the popular clustering method for numerical data. However, the result of the FCM algorithm is not only ineffective to data with outliers and noises but also sensitive due to the initial solution. The possibilistic fuzzy *c*-means (PFCM) algorithm [17] may handle the above shortcomings.

The proposed method named possibilistic multivariate fuzzy weighted *c*-means (PMFWCM) algorithm is based on the concept of multivariate fuzzy *c*-means (MFCM) algorithm [18], which is an extension of FCM and PFCM algorithms. The MFCM algorithm considers the importance of features differently in the membership degree. Besides, the MFCM is also extended with weighted features and applied for interval-valued data [19]. The proposed PMFWCM algorithm has three advantages: (1) adding possibilistic function to reduce the effect of noise data, (2) considering the importance of each attribute in the clustering process, and (3) employing the object–cluster similarity measure [20] for mixed data to calculate the distance between data instances and prototypes. However, there are many input parameters in the model of the proposed PMFWCM algorithm, its clustering result is usually unstable because of these parameters' change. Besides, the clustering result is also sensitive to the random initial centroids. Moreover, the proposed PMFWCM also has a drawback that some existing algorithms already had such as FCM, PFCM algorithms, i.e., the algorithm can be terminated at the local optimal solution. Therefore, the proposed PMFWCM algorithm is combined with metaheuristic approaches to exploit the global optimal solution which is more stable than that of results from the PMFWCM algorithm.

To evaluate the proposed algorithms, some benchmark datasets from the UCI machine learning repository will be collected and implemented. The obtained clustering results will be compared with those of the existing algorithms.

Furthermore, this study will implement the proposed algorithms for real-world applications. The algorithm will be applied to the market segmentation of a soft-drink product in Thailand. The main purpose is to help in clustering customer groups and match them with the types of products. The result of the clustering method will be validated to ensure that the result can be applied in a real dataset of market segmentation. The result is expected to improve the marketing system and give a better experience for marketers.

The remainder of this study is presented as follows. Section 2 shows the related literature review on cluster analysis. Section 3 provides an algorithmic background. Section 4 introduces the proposed methods. Section 5 and Section 6 show the computational results by using the benchmark data sets and real-world case, respectively. Finally, the concluding remakes are drawn in Section 7.

2. Related works

2.1. Cluster analysis

Clustering is known as an unsupervised learning technique usually used to exploit the data structure. It is called an unsupervised approach because clustering is usually performed without available information that is related to the membership of data items to predetermined label. Commonly, clustering is the process of partitioning a given dataset into several distinct clusters in which the objects within a cluster are more similar to each other and dissimilar to those in other clusters [1]. Nowadays, clustering has become a popular and crucial technique since its applications are implemented in many fields.

Generally, clustering algorithms can be divided into two categories, hierarchical clustering and partitioning clustering [21]. With an acknowledgment of the hierarchical clustering, this approach analyzes the dataset into a hierarchy of a group and typically is represented as a dendrogram or tree. The clustering result is determined by cutting the tree at a specified level. The hierarchical methods can be further classified into two categories: (1) agglomerative which uses the bottom-up approaching method and (2) divisive which uses a top-down approach [22]. Regarding the partitioning methods, partitioning clustering is a useful technique for large datasets due to their comparatively low computational cost. There are four major categories of the partitioning approaches, i.e., relocation-based [23], grid-based, subspace-based [24,25] and density-based methods [26].

Clustering itself is not only one specific algorithm. Researchers can design a variety of algorithms based on their approaches to form a cluster effectively. The general concept of clusters includes groups with small intra-cluster distance, dense area of the data space, intervals or particular statistical distributions [27]. Therefore, clustering can be described as a multi-objective optimization problem. Typically, minimizing some measure of dissimilarity in the objects within-cluster and maximizing the dissimilarity of different clusters are the two most common objective functions [1]. Selecting an appropriate clustering algorithm depends on the specific features of each data set.

2.2. Review of clustering for mixed data

Various methods have been proposed for clustering mixed numerical and categorical data. Generally, these clustering methods for mixed data can be divided into two major groups. The first group pays attention to investigating a novel distance similarity for mixed data. Generally, most of the clustering techniques in this group focus on finding an innovative distance metric for categorical attributes while Euclidean distance is usually employed for the numerical attributes. As mentioned in the introduction, the fuzzy *k*-prototype algorithm [16] is the most representative ones. Other innovative distance metric can be listed such as probability-based frequency distance [28], Goodall similarity metric [29], distance hierarchy [30], probabilistic dissimilarity [31], distance-based Rényi entropy and complement entropy [32], object–cluster similarity measure [20], and so on.

The algorithms in the second group employ the pre-processing approach to convert the original-mixed data to single-type data. For instance, the categorical attributes can be transformed into numerical ones and then a clustering method is applied for numerical data. Another way is to convert the numerical attributes into category ones by a discretization method. Then, the clustering method for purely categorical data is applied for the transformed data. The pre-processing approach can be found in the following papers [33–37].

Recently, Ahmad and Khan [38] provided a survey that studied a taxonomy for mixed data clustering based on the research themes. Thereafter, the state-of-the-art review of the clustering algorithms within each research theme was presented. Besides, the pros and cons, as well as major application, and impact areas of the surveyed algorithms were also pointed out in the survey.

3. Algorithmic background

3.1. Fuzzy c-means -based algorithms

3.1.1. Fuzzy c-means algorithm

FCM algorithm [15] is one of the most popular fuzzy clustering algorithms. The FCM algorithm is derived from the concept of fuzzy sets theory. The algorithm can be described as follows. Suppose that there is dataset X which contains n data instances, each data instance is represented by p features. The given dataset is expected to divide into c clusters. Let $U = \left[u_{ij}\right]_{n\times c}$ be the matrix to represent the fuzzy membership degree of object x_i to cluster j, $V = \{v_1, v_2, \ldots, v_c\}$ denote the centroid of c clusters. The FCM algorithm partitions a given dataset into c clusters to minimize the following objective function:

$$\min\left\{J(U,V,X) = \sum_{i=1}^{N}\sum_{j=1}^{c}\left(u_{ij}\right)^m \mathrm{d}\left(x_i, v_j\right)^2\right\}, \tag{1}$$

Subject to

$$u_{ij} \in [0,1]; \sum_{j}^{c} u_{ij} = 1; 1 \le j \le c; 1 < i < n, \tag{2}$$

Herein, Eq. (1) is a distance-based objective function where $\mathrm{d}\left(x_i, v_j\right)$ represents the dissimilarity between object x_i to cluster center v_j. Besides, the weighting exponent, m ($m \in (1,\infty)$), is commonly set as 2 [39,40]. Generally, Euclidean distance is often employed. Thus, $\mathrm{d}\left(x_i, v_j\right)$ is determined as follows:

$$\mathrm{d}\left(x_i, v_j\right) = \sqrt{\sum_{l=1}^{p}(x_{il} - v_{jl})^2}, \tag{3}$$

Eq. (2) shows the constraints of the fuzzy membership degree. The objective function in Eq. (1) and constraint in Eq. (2) are obtained for minimization by using the Lagrange multiplier theorem. To minimize the objective function, the matrix U and centers V are iteratively updated as follows:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d(x_i,v_j)}{d(x_i,v_k)}\right)^{\frac{2}{m-1}}}; 1 \le i \le n; 1 \le j \le c, \quad \text{and} \tag{4}$$

and

$$\mathrm{v}_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}; 1 \le j \le c, \tag{5}$$

The iteration is stopped at iteration Ite if $\left\|V^{Ite} - V^{Ite-1}\right\|_F \le \varepsilon$, where $\|\cdot\|_F$ is Frobenius norm, and ε is a termination criterion. Most of the existing research set $\varepsilon = 10^{-5}$.

The algorithm procedure for FCM algorithm can be described as follows:

Step 1 Predetermine the number of clusters c ($1< c <n$), set the value for m ($m>1$), and set $Ite = 1$ for iteration. Generate the initial cluster centers $V^{[0]}$ for c clusters.

Step 2 Update the fuzzy membership matrix U^{Ite} based on Eq. (4) for iteration Ite.

Step 3: Calculate the cluster center $V^{Ite} = \left\{v_1^t, v_2^t, \ldots, v_c^t\right\}$ by Eq. (5) for iteration Ite.

Step 4: If $\left\|V^{Ite} - V^{Ite-1}\right\|_F \ge \varepsilon$, then $Ite = Ite + 1$ and go to Step 2, else stop.

3.1.2. Multivariate fuzzy c-means algorithm

Regarding the FCM algorithm, the fuzzy memberships are determined based on the distances between objects within-cluster and its centroid where the importance of features is not considered for the definition of the fuzzy membership degree. In fact, the membership values can be restricted to deliberate the influence of features due to its dissimilar dispersion, which can affect the clustering result. To solve this problem, multivariate fuzzy c-means (MFCM) algorithm [18], which is an extension of the FCM method, was proposed. In the MFCM algorithm, the fuzzy membership degrees are calculated based on the inherent information in each feature. Basically, the MFCM generates the membership degrees of data instances based on the characteristic of feature which is beneficial for the subsequent use of clustering results. Like the FCM algorithm, the MFCM algorithm minimizes the distance-based objective functions by finding a set of centroids and a multivariate fuzzy partition through the iteratively updating process. The objective function is illustrated as follows:

$$J_m(U,V;X) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{j=1}^{p} u_{ijk}^m d_{ijk}, \tag{6}$$

where n is the number of data instances with p features, u_{ijk} is the membership degree of data instance x_k to cluster i on dimension j ($u_{ijk} \in [0,1]$), and d_{ijk} is the distance given as $d_{ijk} = (x_{jk} - v_{ij})^2$. Similar to the FCM algorithm, the objective function of MFCM in Eq. (6) is minimized under the following restriction:

$$\begin{aligned}&\sum_{i=1}^{c}\sum_{j=1}^{p} u_{ijk} = 1 \forall k \in \{1, \ldots, n\} \text{ and}\\ &0 < \sum_{j=1}^{p}\sum_{k=1}^{n} u_{ikj} < n \ \forall\ i \in \{1, \ldots, c\},\end{aligned} \tag{7}$$

Summing up the multivariate membership degrees over the dimensions helps to get the membership degrees of data instances to clusters. Like the FCM algorithm, its multivariate version also fails to detect the outliers and noise data. Thus, the MFCM algorithm considers assigning the outliers and noise data to clusters like data instances within clusters.

Similar to the FCM algorithm, the MFCM updates the multivariate fuzzy membership degree and cluster centroids by using the following equations:

$$u_{ijk} = \left(\sum_{h=1}^{c}\sum_{l=1}^{p}\left(\frac{d_{ijk}}{d_{hlk}}\right)^{1/(m-1)}\right)^{-1} \text{ and} \tag{8}$$

$$v_{ij} = \sum_{k=1}^{n}\left(u_{ijk}^m\right) x_{jk} / \sum_{k=1}^{n}\left(u_{ijk}^m\right). \tag{9}$$

3.1.3. Possibilistic multivariate fuzzy c-means algorithm

PFCM algorithm can overcome the existing drawback of the FCM algorithm on outliers. However, the PFCM algorithm cannot provide information about the dimensions where the clusters are overlapped [41]. This information can be useful for the next analysis or process. Thus, the possibilistic multivariate fuzzy c-means (PMFCM) algorithm, which combines the advantage of MFCM and PFCM algorithms, was proposed by [41]. The objective function of PMFCM algorithm is defined as follows:

$$\begin{aligned}J_m(U,T,V;X) = &\sum_{k=1}^{n}\sum_{i=1}^{c}\sum_{j=1}^{p}(au_{ikj}^m + bt_{ik}^{\eta})(v_{ij} - x_{kj})^2\\ &+ p\sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1 - t_{ik})^{\eta}\end{aligned} \tag{10}$$

Herein, the possibilities are defined as typicality values (t). Typicality is a function of both its within-group resemblance and its dissimilarity to other group. The advantage of typicality is expected to reduce the effect of outlier data. The typicality of data instances to clusters t_{ik} are involved in the weighting of the dimension-wise distances between data instances and cluster centroids. Note that it is not necessary to calculate the typicality of data instances to clusters on each dimension because the noise data can lead to a large overall distance to cluster centroids. Different from the MFCM algorithm, the PMFCM algorithm does not subject the sum of multivariate membership degrees over all clusters and variables to a particular instance to be 1, but it still has to restrict the sum over all clusters to a particular instance in each dimension to be 1 to keep the equal distance weight. Thus, the objective function in Eq. (10) is minimized under the following restriction:

$$\sum_{i=1}^{c} u_{ikj} = 1 \ \forall\ k, j \wedge \sum_{k=1}^{n} u_{ikj} > 0 \ \forall\ i, j \wedge \sum_{k=1}^{n} t_{ik} > 0 \ \forall\ i, \tag{11}$$

The membership degrees u_{ikj}, typicality values t_{ik}, and the centroids v_{ij} are updated by using Eq. (12), (13), and (14), respectively.

$$u_{ikj} = \frac{1}{\sum_{l=1}^{c} \left(\frac{(x_{kj}-v_{ij})^2}{(x_{kj}-v_{lj})^2}\right)^{\frac{1}{m-1}}} \quad 1 \le i \le c, 1 \le k \le n, 1 \le j \le p, \tag{12}$$

$$t_{ik} = \frac{1}{1 + \left(\frac{b\sum_{j=1}^{p}(x_{kj}-v_{ij})^2}{\gamma_i p}\right)^{\frac{1}{\eta-1}}} \quad 1 \le i \le c, 1 \le k \le n, \tag{13}$$

$$v_{ij} = \frac{\sum_{k=1}^{n}(au_{ikj}^{m} + bt_{ik}^{\eta})x_{kj}}{\sum_{k=1}^{n}(au_{ikj}^{m} + bt_{ik}^{\eta})} \quad 1 \le i \le c, 1 \le j \le p. \tag{14}$$

3.2. Clustering approaches for mixed attributes

3.2.1. Fuzzy k-prototypes algorithm

Regarding the clustering methods for mixed numerical and categorical data, fuzzy k-prototype [16] is one of the most popular ones. This algorithm aims to partition a given mixed dataset with n data instances, p attributes to c clusters. The data instances are described by d_u numerical attributes and d_c categorical attributes, where (d_c + d_u = p). Thus, the prototype of cluster j (Q_j) consists of two parts: (1) q_{jl}^{r} is the centroid for numerical attributes, and (2) q_{jl}^{s} is the centroid for categorical attributes. The cost function of fuzzy k-prototype is shown as follows:

$$J = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{\alpha} d(x_i, Q_j), \tag{15}$$

where $U = (u_{il})_{nk}$ is the partition matrix, α is fuzziness coefficient. $d\left(x_i, Q_j\right)$ is the distance from an instance x_i to cluster j.

$$d\left(x_i, Q_j\right) = \sum_{l=1}^{d_u} w_l (x_{il}^{r} - q_{jl}^{r})^2 + \sum_{j=1}^{d_c} d(x_{ij}^{s}, q_{jl}^{s}), \tag{16}$$

The distance $d\left(x_i, Q_j\right)$ consists of two parts: distance of numerical attributes $\sum_{l=1}^{d_u} w_l (x_{il}^{r} - q_{jl}^{r})^2$ and distance of categorical ones $\sum_{j=1}^{d_c} d(x_{ij}^{s}, q_{jl}^{s})$. Herein, w_l is the significance of the numerical attribute l. The fuzzy partition matrix is defined by

$$u_{ij} = \left(\sum_{z=1}^{c}\left(\frac{d\left(x_i, Q_j\right)}{d\left(x_i, Q_z\right)}\right)^{1/(\alpha-1)}\right)^{-1}, \tag{17}$$

The centroid for numerical and categorical attributes are updated by Eq. (18) and (19), respectively.

$$q_{jl}^{r} = \frac{\sum_{i=1}^{n} u_{ij}^{\alpha} x_{il}^{r}}{\sum_{i=1}^{n} u_{ij}^{\alpha}}, \tag{18}$$

$$q_{jl}^{s} = \frac{a_{jl}^{1}}{w_{jl}^{1}} + \frac{a_{jl}^{2}}{w_{jl}^{2}} + \cdots + \frac{a_{jl}^{k}}{w_{jl}^{k}} + \cdots + \frac{a_{jl}^{t}}{w_{jl}^{t}}, \tag{19}$$

where

$$w_{jl}^{c} = \sum_{i=1}^{n} \gamma(x_{il}^{s}), \tag{20}$$

$$\gamma(x_{il}^{s}) = \begin{cases} \dfrac{u_{ij}^{\alpha}}{\sum_{i=1}^{n} u_{ij}^{\alpha}}, & a_{jl}^{c} = x_{il}^{s} \\ 0, & a_{jl}^{c} \ne x_{il}^{s}. \end{cases} \tag{21}$$

The procedure of the fuzzy k-prototype algorithm is presented as follows:

Step 1: Parameters setting: maximum iteration T, number of clusters c, fuzziness coefficient α, stopping criteria ε. Randomly select c initial prototypes $Q^{(Ite)}$ and set iteration $Ite = 0$.

Step 2: Compute the fuzzy partition matrix $U^{(Ite)}$ by Eq. (17).

Step 3: Update the prototype $Q^{(Ite+1)}$ using Eqs. (18)–(21).

Step 4: If $\left\|J^{Ite} - J^{Ite-1}\right\|_F \ge \varepsilon$, set $Ite = Ite+1$, $\max T = \max(T-1)$ and then go to Step 2; otherwise stop.

3.2.2. Subspace clustering

The subspace clustering aims to determine the clusters based on characteristics of the feature subsets [25]. Jia and Cheung [19] proposed a subspace clustering algorithm for mixed data which employed the ideas of k-means-type methods and object–cluster similarity for distance measure. Besides, the proposed method can detect the number of clusters for the datasets with an unknown number of clusters. The algorithm is described as follows. Suppose there is a given dataset X with n objects, $\{x_1, x_2, \ldots, x_n\}$, which is divided into c distinguish clusters ($C_1, C_2, \ldots, C_c$). Each mixed data object x_i is represented with p different attributes $\{A_1, A_2, \ldots, A_p\}$, which consists of d_c categorical attributes $\{A_1^c, A_2^c, \ldots, A_{d_c}^c\}$ and d_u numerical attributes $\{A_1^u, A_2^u, \ldots, A_{d_u}^u\}$, d_c + d_u = p. Thus, x_i can be denoted as $[x_i^{c^T}, x_i^{u^T}]^T$ with $x_i^c = (x_{i1}^c, x_{i2}^c, \ldots, x_{id_c}^c)^T$ and $x_i^u = (x_{i1}^u, x_{i2}^u, \ldots, x_{id_u}^u)^T$. Here, x_{ir}^u ($r = 1, 2, \ldots, d_u$)belongs to which is the set of real numbers, and x_{ir}^c ($r = 1, 2, \ldots, d_c$)belongs to $dom(A_r^c)$ where $dom(A_r^c) = a_{r1}, a_{r2}, \ldots, a_{rm_r}$ has m_r elements including all the possible values that can be selected by attribute A_r^c. The proposed method of Jia and Cheung [19] aims to optimize the objective function J*:

$$\mathrm{J}^{*} = \arg\max\left[\sum_{j=1}^{c}\sum_{i=1}^{n} q_{ij} d\left(x_i, C_j\right)\right], \tag{22}$$

where $d(x_i, C_j)$ is the distance between instance x_i and cluster C_j, and $Q = (q_{ij})$ is an partition matrix sized n x c, which is subject to:

$$\sum_{j=1}^{c} q_{ij} = 1, \ \text{and}\ 0 < \sum_{j=1}^{c} q_{ij} < n, \tag{23}$$

with

$$q_{ij} \in [0, 1], \quad i = 1, 2, \ldots, n, j = 1, 2, \ldots, c, \tag{24}$$

The object–cluster similarity for mixed numerical and categorical attributes can be simply defined as:

$$d\left(x_i, C_j\right) = \frac{1}{d_f}\left\{\sum_{r=1}^{d_c} d\left(x_{ir}^c, C_j\right) + d\left(x_i^u, C_j\right)\right\}, \tag{25}$$

with

$$d\left(x_{ir}^c, C_j\right) = \frac{\Psi_{A_r^c = x_{ir}^c}(C_j)}{\Psi_{A_r^c \neq NULL}(C_j)}, \tag{26}$$

and

$$d\left(x_i^u, C_j\right) = \frac{\exp(-0.5\mathrm{Dis}\left(x_i^u, c_j\right))}{\sum_{t=1}^{k}\exp(-0.5\mathrm{Dis}\left(x_i^u, c_t\right))}. \tag{27}$$

where $d_f = d_c+1$. $\Psi_{A_r^c = x_{ir}^c}(C_j)$ measures the number of data instances within cluster C_j that contain the value x_{ir}^c for feature A_r^c, NULL means the empty, $\Psi_{A_r^c \neq NULL}(C_j)$ represents the number of data instances within cluster C_j that contain the feature A_r^c. The centroid of all numerical features in cluster C_j is illustrated by c_j, and $Dis(.)$ is denoted for a distance function.

3.3. Metaheuristics

Meta-heuristic approaches are considered to be high-level strategies for complex problems by coordinating collaboration between other search methods, including heuristics and traditional search techniques [42]. Meta-heuristics can be formally defined as an iterative process that intelligently combines different concepts to explore and exploit search spaces to effectively find approximately optimal solutions [43]. Meta-heuristics can be used when the problems cannot be solved by classical optimization techniques in polynomial time. Although the results can be found more efficiently and efficiently by meta-heuristic approaches, there is no guarantee that the global optimal solution can be found. The next sub-section will describe the meta-heuristic approach which will be used in this project.

3.3.1. Genetic algorithm (GA)

Genetic algorithm (GA), which was first proposed by Holland [44], is one of the most popular meta phenomena of natural evolution and population genetics. The algorithm was first applied in a real case in 1988 [45]. Commonly, GA appropriates for solving complex optimization problems. The general procedure of GA can be described as follows. First, the chromosome is encoded to represent the characteristics of solution using genotype. The initial population which contains the number of chromosomes is generated in the initialization stage and then the genetic operation including selection, crossover, and mutation process is repeated iteratively. To select the chromosome for the genetic operation, each chromosome is evaluated by the fitness function. The selection process aims to select better chromosomes to reproduce the next generation. There are two popular selection methods, i.e., roulette wheel and tournament. In the roulette wheel method, selecting a chromosome for generating the next generation is based on the proportion of its fitness value. The chromosome, which has better fitness value, will have higher chance to be selected. In contrast, the tournament method is a process to run a few tournaments among a few chromosomes that are randomly chosen from the population. Thereafter, the winner chromosome of each tournament, which is the one with the best fitness, is chosen for the next genetic operation. After selecting the good chromosomes, crossover and mutation processes are performed sequentially. The offspring chromosomes for the next generation are formed. The genetic operation is repeated iteratively until meeting the stopping criteria, which may be set by the number of generations. Then, the final solution can be obtained. There are many examples of different types of GA combined with clustering methods [46,47] to avoid the local optimal solutions for clustering [48].

3.3.2. Particle Swarm Optimization Algorithm (PSO)

PSO [49] is an evolutionary method based on the swarm theory which is inspired by the social behavior of bird flocking or fish schooling. PSO algorithm is used to solve the optimization problems for both discrete and continuous variables, as well as single and multi-objective models. Like GA, PSO also randomly generates the initial population and update the solutions iteratively. However, there are no genetic operators such as crossover and mutation in the PSO procedure. Herein, each solution is illustrated as a particle in which each particle will move around the search space to find the most promising area or the best solution. Each particle in the population is identified by two factors, i.e., velocity and location, which are updated iteratively in each iteration by moving around in the search space. The particles are updated based on two best values: Gbest (global best) and Pbest (particle best). Pbest is a local best-known solution obtained by the specific particle, while Gbest is the overall best solution obtained by the swarm. The current optimal particles in each iteration are determined to make a direction in the search space for potential solutions.

There are two advantages of PSO, as compared to GA, i.e., simpler to implement, and fewer parameters to justify. Thus, PSO has been used in various fields such as evolutionary computing, optimization, neural network training, control system design as well as other fields that GA can be applied [50].

3.3.3. Sine Cosine Algorithm (SCA)

The sine cosine algorithm (SCA) is a new meta-heuristic approach proposed by Seyedali Mirjalili in 2016 [51]. SCA uses a simple concept but can achieve a very impressive result in solving optimization problems. Generally, SCA includes two phases. The initialization phase randomly generates solutions in the search space. Next, sine and cosine functions are used to update the locations of the initial solutions in the exploitation phase. By employing the trigonometric functions, i.e., sine and cosine functions, the solution will fluctuate outwards or towards the optimal solution. Moreover, there are four parameters in updating formulations used to decide the direction, distance, trade-off, and switch from cosine to sine or sine to cosine. Unlike PSO, SCA only records the best solution in the previous iteration and decides the moving distance by generating a random number without calculating the difference between solutions. Similar to other metaheuristic approaches such as GA and PSO, SCA also faces the major drawbacks, i.e., its accuracy and convergence are affected by the calibration and randomness of some internal parameters.

4. Methodology

The proposed Possibilistic Multivariate Fuzzy Weighted c-means (PMFWCM) algorithm, which is a combination of the PMFCM and MFCM algorithms with weighting, is first introduced in this section. The original versions of PMFCM and MFCM algorithms are applied for numerical data. To deal with the mixed data, the new dissimilarity measure is employed to calculate the distance of the categorical attributes. Thereafter, the combination of PMFWCM and metaheuristic methods, such as SCA, GA, and PSO, to exploit the global optimal solution are presented. The detail of each algorithm is described in the following sub-section.

4.1. Possibilistic multivariate fuzzy weighted c-means (PMFWCM) algorithm

The proposed PMFWCM algorithm for mixed numerical and categorical attributes is described in this sub-section. First, the

objective of PMFWCM algorithm is shown in the following function:

$$J_{m,\eta}(U,T,V,W;X)=\sum_{k=1}^{n}\sum_{i=1}^{c}\sum_{j=1}^{p}[a(\frac{u_{ikj}}{w_{ikj}})^{m}+bt_{ik}^{\eta}]d_{ijk}^{2}+p\sum_{i=1}^{c}\gamma_i\sum_{k=1}^{n}(1-t_{ik})^{\eta} \quad (28)$$

Subject to

$$\sum_{i=1}^{c}u_{ikj}.w_{ikj}=1\ \forall k,j\wedge\sum_{k=1}^{n}u_{ikj}.w_{ikj}>0\ \forall i,j\wedge\sum_{k=1}^{n}t_{ik}>0\ \forall i, \quad (29)$$

To minimize the objective function, the $J_{m,\eta}$, the fuzzy membership degrees, typicality, and prototypes are updated iteratively as follows:

$$u_{ikj}=\frac{1}{\sum_{h=1}^{c}\sum_{l=1}^{p}(\frac{d_{ikj}}{d_{hkl}})^{\frac{1}{m-1}}(\frac{w_{hkl}}{w_{ikj}})^{\frac{m}{m-1}}},\quad 1\le i\le c,1\le k\le n,1\le j\le p, \quad (30)$$

$$t_{ik}=\frac{1}{1+(\frac{b\sum_{j=1}^{p}(x_{kj}-v_{ij})^{2}}{\gamma_i p})^{\frac{1}{\eta-1}}},1\le i\le c,1\le k\le n, \quad (31)$$

where:

$$\gamma_i=K\frac{\sum_{k=1}^{n}(\frac{u_{ikj}}{w_{ikj}})^{m}d_{ikj}^{2}}{\sum_{k=1}^{n}(\frac{u_{ikj}}{w_{ikj}})^{m}},\quad 1\le i\le c,1\le j\le p, \quad (32)$$

$$w_{ikj}=\frac{(u_{ikj}^{m}d_{ikj})^{\frac{1}{m+1}}}{\sum_{h=1}^{c}\sum_{l=1}^{p}(u_{hkl}^{m}d_{hkl})^{\frac{1}{m+1}}u_{hkl}},\quad 1\le i\le c,1\le k\le n,1\le j\le p, \quad (33)$$

The cluster centroids or prototypes of mixed data consist of two parts: centroid for numerical attributes and centroid for categorical attributes which are updated as follows:

(1) For numerical attributes:

$$v_{ij}=\frac{\sum_{k=1}^{n}[a(\frac{u_{ikj}}{w_{ikj}})^{m}+bt_{ik}^{\eta}]x_{kj}}{\sum_{k=1}^{n}[a(\frac{u_{ikj}}{w_{ikj}})^{m}+bt_{ik}^{\eta}]}\quad 1\le i\le c,1\le j\le p, \quad (34)$$

(2) For categorical attributes: the proposed algorithm employs the frequency-based method to update centroid.

Similar to the cluster centroids, the distance d_{ikj} in the proposed PMFWCM is also separated into two parts to calculate.

(1) For numerical attributes: d_{ikj} is calculated based on the Euclidean distance

$$d_{ikj}=(x_{jk}-v_{ij})^{2}, \quad (35)$$

(2) For categorical data: d_{ikj} is calculated based on the object–cluster similarity

$$d_{ikj}=d(x_{kr},C_j)=\frac{\Psi_{a=x_{kr}}(C_j)}{\Psi_{a\neq NULL}(C_j)}. \quad (36)$$

where $\Psi_{a=x_{kr}}(C_j)$ counts the number of objects in cluster C_j the have value x_{kr} for attribute A_r $\Psi_{a\neq NULL}(C_j)$ is the number of objects in cluster C_j that have the attribute A_r.

The procedure of the proposed PMFWCM is illustrated as follows:

Step 1: Parameters setting: number of clusters (c), a, b, m and η, number of iterations (T), and terminal condition (ε).

Step 2: Initialize membership degree (u_{ikj}) and typical value (t_{ik}) where $i=1,2,\ldots,$ c (ith cluster), $k=1,2,\ldots,n$(kth data object), $j=1,2,\ldots,p$(jth feature)

Step 3 Initialize the weight (w_{ikj}). Set iteration $Ite=1$.

Step 4: Determine the centroid (v_{ij}) in each cluster based on Eq. (34) for numerical attributes, and frequency-based method for categorical attributes.

Step 5 Calculate the distance between the data objects to cluster centers using Eqs. (35)–(36)

Step 6: Compute the penalty (γ_i) using Eq. (32)

Step 7: Update the membership degree (u_{ikj}) and the typical value (t_{ik}) and the weight (w_{ikj}) using Eqs. (30)–(31) and (33).

Step 8: Compute the objective function J^{Ite}

Step 9: If $|J^{Ite}-J^{Ite-1}|\le\varepsilon$, or $Ite=T$, then stop; otherwise, $Ite=Ite+1$ and go to Step 4.

4.2. Metaheuristics-based clustering

4.2.1. Sine cosine algorithm-based clustering

Since there are many parameters in the mathematical model of the proposed PMFWCM algorithm, the clustering result is usually unstable because of these parameters' change. Besides, the random initial centroids also affect the clustering result. Moreover, the proposed PMFWCM also exists a drawback that is like the FCM algorithm, i.e., the algorithm can be terminated at the local optimal solution. Thus, this study proposed a SCA-based PMFWCM algorithm to overcome these problems. Herein, the SCA is used to exploit both the initial centers and parameters of the proposed PMFWCM algorithm. This study uses global compactness as the fitness function in the meta-heuristic and calculates the accuracy to verify the result. The global compactness G which should be as small as possible is be defined as follows.

$$G=\sum_{i=1}^{c}\frac{\sum_{k}^{n}u_{ik}d(x_k,v_i)}{\sum_{k}^{n}u_{ik}}, \quad (37)$$

where n is the number of data points, c is the number of clusters and u_{ik} is the membership degree.

SCA is a population-based evolutionary algorithm. The algorithm can increase the probability to get the optimal result by generating a number of random results through the optimization steps. To obtain the optimization search, the population position is updated by using sine and cosine functions as follows:

$$X_i^{(Ite+1)}=X_i^{Ite}+r_1\times\sin(r_2)\times\left|r_3P_i^{Ite}-X_i^{Ite}\right|,\ r_4<0.5\text{ and} \quad (38)$$

$$X_i^{(Ite+1)}=X_i^{Ite}+r_1\times\cos(r_2)\times\left|r_3P_i^{Ite}-X_i^{Ite}\right|,\ r_4\ge0.5 \quad (39)$$

where X_i^{Ite} is the position of the ith attribute at iteration Ite, P_i^{Ite} is the optimal position at iteration Ite, r_1, r_2, r_3 and r_4 are random parameters. Herein, r_2 is set in random between 0 to 2π while r_3 is set in random between 0 to 2 and r_4 is set in random between 0 to 1.r_1 is the controlled parameter to balance exploration which is altered adaptively using the following function:

$$r_1=a-Ite\frac{a}{T}, \quad (40)$$

where *IteandT* are the current and maximum iteration, respectively, and a is a user constant.

Solution representation the proposed SCA-PMFWCM algorithm uses the cluster centroids as the solution representation. Herein, the cluster centroids consist of two parts: centroids for numerical attributes and centroids for categorical attributes. The solution representative is illustrated in Fig. 1.

Objective function: the proposed SCA-PMFWCM uses global compactness G in Eq. (37) as the fitness function.

The following steps are used to describe the procedure of SCA-PMFWCM algorithm:

Step 1: Parameters setting including population size, a value in r_1 and the maximum iteration T.

Step 2 Initialization: run PMFWCM to get the cluster centroids for the initial population. Parameters of the PMFWCM algorithm are randomly generated. Set $Ite = 1$ for iteration.

Step 3: Compute fitness value using Eq. (37)

Step 4: Find the optimal solution P_i^{Ite}.

Step 5: Calculate $r_1$by Eq. (40).

Step 6: Randomly generate r_2, r_3 and r_4.

Step 7: Update the position of the population using Eqs. (38) and (39).

Step 8: Run the PMFWCM algorithm to evaluate the fitness and obtain the new cluster centroids of the next generation.

Step 9: Stop the algorithm if the terminate condition is met; otherwise, back to Step 4.

Step 10: Obtain the optimal solution from the SCA-PMFWCM algorithm after the final generation.

4.2.2. Genetic algorithm-based clustering

In this study, the proposed GA based PMFWCM algorithm (GA-PMFWCM) encodes the chromosome representation like the solution representation of the SCA-PMFWCM algorithm. Besides, global compactness G is also selected as a fitness function.

The steps of GA-PMFWCM algorithm are presented as follows:

Step 1: For GA, parameters setting including numbers of chromosomes (P) and generations (G), number of clusters (c), crossover rate, and mutation rate. Also set up the parameters for the PMFWCM algorithm according to Section 4.1.

Step 2 Initialization: randomly generate the initial population. Use the PMFWCM procedure to cluster the data.

Step 3: Compute the fitness of each chromosome using Eq. (37).

Step 4: Selection: employ roulette wheel technique to select the chromosomes into the crossover pool.

Step 5: Genetic operation: Implement crossover and mutation to reproduce the offspring using the following equations [52]:

Crossover:

$$O_1 = \alpha \times P_1 + (1-\alpha) \times P_2 \tag{41}$$

$$O_2 = \beta \times P_1 + (1-\beta) \times P_2 \tag{42}$$

where P_1, P_2 are the two parent chromosomes which are selected from the population for crossover, O_1, O_2 are the offsprings after crossover, and α, β are two random numbers uniformly distributed in the range of (0,1).

Mutation:

$$Var_i = Var_i + s_i \times r_i \times q_i \tag{43}$$

where $s_i \in \{-1, +1\}$ and is determined randomly. $r_i = r \times range_i$ with $r \in \left[10^{-6}, 0.1\right]$, $range_i$ depends on the problem. $q_i = 2^{-uk}$ with $u \in [0, 1]$ and $k \in \{4, 5, \ldots, 20\}$. All s, r, and k in Eq. (39) are determined by user.

Step 6: Run the PMFWCM algorithm to evaluate the fitness values and obtain the new cluster centroids for the next generation.

Step 7: Terminate if the stopping condition is met; otherwise, back to Step 3.

4.2.3. Particle swarm optimization algorithm-based clustering

The PSO algorithm is also a powerful method to efficiently find the optimal solution by updating the particles' location and velocity based on two best values: Gbest (global best) and Pbest (particle best). The velocity (V) and location (X) are updated as follows:

$$V_i^{(T+1)} = w \times V_i^{(T)} + r_1 c_1 \left(X_{Pbest} - X_i^T\right) + r_2 c_2 \left(X_{Gbest} - X_i^T\right), \tag{44}$$

$$X_i^{T+1} = X_i^T + V_i^{(T+1)}, \tag{45}$$

where w is inertial coefficient, r_1 and r_2 are two random numbers ($0 \le r_1, r_2 \le 1$), and c_1 and c_2 are acceleration coefficients.

The proposed PSO based PMFWCM algorithm (PSO-PMFWCM) uses similar solution representation and fitness function with SCA and GA based PMFWCM algorithms.

The steps of PSO-PMFWCM algorithm:

Step 1 Parameters setting for PSO algorithm: numbers of particles (S), maximum number of iterations (T), w, r_1, r_2, c_1 and c_2, and number of clusters (c). Also, setup the parameters for the PMFWCM algorithm.

Step 2: Set the current iteration $Ite = 0$. Generate the initial locations randomly. Using the PMFWCM procedure to cluster the data.

Step 3: Compute the fitness values of particles using Eq. (37). Find the P_{best} and G_{best}.

Step 4: Update the velocity and location using Eq. (44–45).

Step 5: Run the PMFWCM algorithm to evaluate fitness values. Update the P_{best} and G_{best}.

Step 6: If $Ite = T$, then stop; otherwise, increase $Ite = Ite + 1$ and return to Step 4.

Step 7: Obtain the optimal solution of the PSO-PMFWCM.

5. Computational results

5.1. Data sets

This research uses some benchmark datasets, which consist of numerical data and mixed data, collected from the UCI machine learning repository (https://archive.ics.uci.edu/ml/index.php) to validate the clustering performance of the proposed algorithms. The benchmark datasets are shown in Table 1, where N is number of data instances, d_u is number of numerical attributes, d_c is number of categorical attributes, and k is number of clusters (pre-determined).

5.2. Data preprocessing

Data preprocessing plays a crucial role in data mining. Each feature has a different scale. If the original data is used for clustering, the results of the data are easily affected by large-scale features. Therefore, data that has not been pre-processed can have misleading results. In this study, a normalization method is

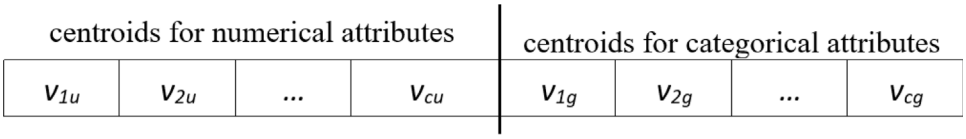

Fig. 1. Solution representation of the SCA-PMFWCM algorithm.

Table 1
Benchmark datasets.

Data features	Datasets	N	d_u	d_c	k
Numerical data	Iris	150	4		3
	Wine	178	13		3
	Haberman	306	3		2
	Glass	214	9		7
	Pima Indians	768	8		2
Mixed data	Heart	303	7	6	2
	Credit	653	9	6	2
	German	1000	13	7	2
	Dermatology	366	33	1	6
	Adult	45222	8	6	2
	Zoo	101	2	15	7
	Flags	194	10	20	8
	Hepatitis	155	6	14	2

Table 2
Parameter setting in each method.

Method	Parameters	Level 1	Level 2	Level 3
SCA	α	1.5	2	3
GA	c_r	0.75	0.9	–
	m_r	0.01	0.05	0.1
PSO	w	0.4	0.8	–
	c_1	0.5	1	2
	c_2	0.5	1	2

used to preprocess the data to avoid deviation. This method scales the features of the data to a range of 0 to 1. The normalized data is defined as:

$$X_i^{NEW} = \frac{X_i - X_i^{min}}{X_i^{max} - X_i^{min}}, \tag{46}$$

where X_i^{NEW} represents normalization data. X_i^{max} and X_i^{min}stand for the maximum and minimum values of the ith feature in the original data, respectively.

5.3. Parameter setting

To find the optimal combination of tuning parameters for SCA, GA, and PSO algorithms, this study employs the method of general factorial design. Table 2 shows the parameter setting in each method. For SCA, parameter r_1 from Eq. (39) is controlled by constant a and number of iterations. Herein, the constant a's levels are determined as 1.5, 2, and 3, respectively. Regarding GA, the crossover rate (c_r) and mutation rate (m_r) are set in range [0,1]. Generally, the crossover rate is high while the probability of mutation is low. Thus, there are two levels set for crossover rate, i.e., 0.75 and 0.9, respectively. In contrast, the mutation rate is set at 0.01, 0.05 and 0.1. In terms of the PSO algorithm, three controlled parameters are employed. Two acceleration constants c_1 and c_2 are determined as 0.5, 1, and 2. The inertia weight (w) is set as 0.4, and 0.8 at the beginning and reduced until 0.9 in these researches. Besides, the number of iterations, which is not considered performing in the factorial design, is set as 100 for SCA, GA, and PSO algorithms.

To evaluate the performance of the proposed algorithms, clustering accuracy (*Acc*) is used. Accuracy indicates the correct percentage of clustering compared with the true label. The clustering accuracy is defined as follows [53]:

$$Acc\,(C, T) = \frac{1}{n}\sum_{i=1}^{c} a_i. \tag{47}$$

where $C = c_1, c_2, \ldots, c_c$ and $T = t_1, t_2, \ldots, t_c$ are the clustering results and the true class labels, respectively. a_i is the maximum number of objects that have the same class label with the true lass label, $a_i = max_{j=1}^{c} n_{i,j}$ with $n_{i,j}$ is the number of common objects in cluster c_j and t_j.

Table 3
Average accuracy and standard deviation of SCA-PMFWCM algorithm on different tuning parameters.

Tuning parameter	Constant a	1.5	2	3
Heart	Average	0.825	**0.834**	0.819
	SD	0.016	0.017	0.022
Credit	Average	0.779	**0.791**	0.783
	SD	0.033	0.028	0.028
German	Average	0.712	**0.718**	0.713
	SD	0.027	0.027	0.026
Dermatology	Average	0.801	**0.808**	0.803
	SD	0.032	0.028	0.031
Adult	Average	0.784	0.794	**0.797**
	SD	0.022	0.020	0.017
Zoo	Average	0.895	**0.902**	0.897
	SD	0.019	0.013	0.015
Flags	Average	0.602	**0.608**	0.604
	SD	0.021	0.012	0.011
Hepatitis	Average	0.838	**0.844**	0.837
	SD	0.021	0.019	0.022

Table 3 shows the average *Acc* and its standard deviation of SCA-PMFWCM algorithm on different tuning parameters. According to Table 3, the best tuning parameters of the SCA-PMFWCM algorithm for constant a is 2 for Heart dataset, Credit dataset, German dataset, Dermatology dataset, Zoo dataset, Flags dataset, and Hepatitis dataset, and 3 for Adult dataset.

Similarly, Table 4 shows that the best tuning parameters of GA-PMFWCM algorithm for crossover rate and mutation rate are 0.75, and 0.05 for Heart dataset, Credit dataset, German dataset, and Dermatology dataset; 0.75, and 0.01 for Zoo dataset, Flags dataset, and Hepatitis dataset; and 0.9, 0.05 for Adult dataset.

Regarding the PSO-PMFWCM algorithm, the average *Acc* and its standard deviation in Table 5 exhibit that the best tuning parameters for inertia weight, acceleration constant c_1, and acceleration constant c_2 are 0.8, 1, and 2, respectively for Heart dataset, Credit dataset, German dataset, and Dermatology dataset, while these parameters are set as 0.8, 0.5, and 2 respectively for Zoo dataset, Flags dataset, and Hepatitis dataset, and 0.4, 2, 0.5 for Adult dataset.

Moreover, a statistical test is also conducted for each method to evaluate the effect of tuning parameters. The clustering results, i.e., accuracy corresponding to each tuning parameter of each algorithm, are used to perform the statistical test. These statistical tests use t-test procedure with a 95% significance level. Different tuning parameter in these two methods does not significantly influence the result.

Table 4
Average accuracy and standard deviation of GA-PMFWCM algorithm on different tuning parameters.

Tuning parameter	c_r	0.75			0.9		
	m_r	0.01	0.05	0.1	0.01	0.05	0.1
Heart	Average	0.831	**0.834**	0.829	0.826	0.827	0.826
	SD	0.022	0.017	0.023	0.022	0.022	0.021
Credit	Average	0.789	**0.790**	0.784	0.787	0.783	0.785
	SD	0.035	0.033	0.037	0.033	0.032	0.040
German	Average	0.709	**0.717**	0.706	0.710	0.713	0.708
	SD	0.028	0.022	0.028	0.024	0.027	0.029
Dermatology	Average	0.796	**0.804**	0.801	0.797	0.797	0.799
	SD	0.032	0.031	0.031	0.037	0.033	0.036
Adult	Average	0.794	0.795	0.792	0.795	**0.796**	0.793
	SD	0.016	0.015	0.014	0.014	0.012	0.016
Zoo	Average	**0.902**	0.897	0.896	0.899	0.897	0.895
	SD	0.011	0.014	0.011	0.014	0.014	0.012
Flags	Average	**0.607**	0.601	0.604	0.602	0.603	0.601
	SD	0.012	0.012	0.013	0.013	0.014	0.015
Hepatitis	Average	**0.841**	0.836	0.835	0.838	0.836	0.834
	SD	0.022	0.024	0.025	0.022	0.024	0.024

Table 6 shows the *p*-value of the statistical test for the SCA-PMFWCM. The result shows that tuning parameters do not have significant influence on the results except Heart dataset.

Regarding the GA-PMFWCM algorithm, Table 7 shows that tuning parameters do not have significant influence on the results for all datasets.

Similarly, Table 8 shows that tuning parameters are only significant influence on the performance of the PSO-PMFWCM algorithm for Hepatitis dataset. Then, the optimal combinations of tuning parameters are summarized in Table 9.

5.4. Experimental results and analysis

First, the comparison of the proposed PMFWCM algorithm with some benchmark algorithms for numerical data is conducted. Herein, the clustering algorithms for numerical attributes reviewed in Section 2 are selected as the benchmark ones. Table 10 shows the comparison results in *Acc* and its standard deviation (*SD*).

As shown in Table 10, the clustering results of the proposed PMFWCM algorithm are the best in all datasets. In the next step, the PMFWCM algorithm will be developed and implemented in mixed-attribute dataset.

To analyze the performance of each method for a mixed attribute dataset, 30 independent runs are performed using optimal combinations of tuning parameters. The clustering performance of the proposed algorithms is compared with some benchmark methods, i.e., fuzzy *k*-prototype [16], subspace clustering for mixed data (WOCIL algorithm) [20], and sequential ensemble clustering generation algorithm (SECG algorithm) [53]. The stopping criterion for all algorithms is the number of iterations, which is set as 100 times. The results are summarized in Table 11, where *c. time* is computational time (in seconds).

Table 11 shows that the SCA-PMFWCM, GA-PMFWCM, and PSO-PMFWCM algorithms are more accurate than the PMFWCM and other benchmark algorithms on most of the tested datasets except the Flags dataset where the SECG algorithm the best one on this data. Furthermore, combining the PMFWCM with meta-heuristics provides better consistency because of smaller *SD* as well as better accuracy. The PMFWCM has the lowest *Acc* compared to the meta-heuristic based algorithms, yet the difference is not significant. However, the PMFWCM algorithm is still better than the two benchmark algorithms, i.e,. the fuzzy *k*-prototype and WOCIL algorithms. In the comparison with the SECG algorithm, the PMFWCM algorithm is only better on 3 datasets, i.e., Heart, German, and Zoo datasets while the SECG algorithm performed better on the 4 remaining datasets.

For better visualization, Fig. 2 shows the comparison among seven algorithms in terms of the average accuracy. Fig. 2 can be inferred the minimum or the worst accuracy is fuzzy *k*-prototype algorithm, and the algorithm which reaches maximum accuracy in overall is SCA-PMFWCM.

The SCA-PMFWCM algorithm provides better *Acc* for 6 datasets, the GA-PMFWCM gives better *Acc* for 1 dataset, and the PSO-PMFWCM gives better *Acc* for 2 datasets. In conclusion, the SCA-PMFWCM is appropriate for middle and big size datasets, the GA-PMFWCM is appropriate for middle size datasets, and the PSO-PMFWCM is suitable for small size datasets.

In terms of computational time, the proposed SCA-PMFWCM, GA-PMFWCM, and PSO-PMFWCM algorithms take too much time to implement as compared with the other ones. Besides, the SCA-PMFWCM algorithm is faster than the GA-PMFWCM and PSO-PMFWCM algorithms. It shows that the SCA-PMFWCM algorithm is more efficient than two other meta-heuristic algorithms. Thus, it can be concluded that the SCA-PMFWCM has better

Table 5
Average accuracy and standard deviation of PSO-PMFWCM algorithm on different tuning parameters.

Tuning parameter	w	0.4									0.8								
	c_1	0.5			1			2			0.5			1			2		
	c_2	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2
Heart	Average	0.830	0.828	0.824	0.830	0.824	0.820	0.827	0.824	0.825	0.826	0.823	0.824	0.822	0.823	**0.832**	0.826	0.824	0.826
	SD	0.023	0.023	0.023	0.022	0.024	0.022	0.024	0.026	0.023	0.022	0.023	0.022	0.026	0.022	0.020	0.023	0.019	0.025
Credit	Average	0.781	0.776	0.785	0.773	0.786	0.771	0.777	0.780	0.774	0.771	0.770	0.778	0.772	0.778	**0.790**	0.777	0.788	0.773
	SD	0.032	0.027	0.029	0.032	0.031	0.027	0.033	0.031	0.033	0.033	0.029	0.029	0.031	0.031	0.028	0.029	0.027	0.027
German	Average	0.704	0.696	0.710	0.702	0.699	0.701	0.691	0.704	0.711	0.702	0.703	0.710	0.703	0.707	**0.715**	0.701	0.695	0.700
	SD	0.026	0.028	0.028	0.030	0.029	0.034	0.027	0.032	0.028	0.030	0.029	0.030	0.026	0.029	0.028	0.030	0.027	0.026
Dermatology	Average	0.790	0.788	0.799	0.785	0.804	0.789	0.791	0.795	0.805	0.804	0.802	0.795	0.798	0.795	**0.806**	0.805	0.800	0.790
	SD	0.036	0.035	0.036	0.031	0.032	0.034	0.031	0.031	0.035	0.031	0.036	0.033	0.031	0.035	0.031	0.031	0.035	0.033
Adult	Average	0.785	0.786	0.789	0.780	0.783	0.780	**0.790**	0.788	0.789	0.780	0.783	0.784	0.787	0.786	0.778	0.787	0.787	0.781
	SD	0.021	0.020	0.019	0.024	0.022	0.022	0.019	0.023	0.018	0.022	0.023	0.021	0.020	0.021	0.019	0.021	0.021	0.023
Zoo	Average	0.897	0.894	0.897	0.896	0.896	0.892	0.893	0.895	0.894	0.895	0.891	**0.903**	0.892	0.899	0.889	0.893	0.895	0.893
	SD	0.019	0.018	0.019	0.017	0.018	0.018	0.019	0.020	0.020	0.015	0.021	0.018	0.020	0.018	0.016	0.023	0.018	0.018
Flags	Average	0.605	0.608	0.602	0.608	0.604	0.602	0.606	0.604	0.609	0.600	0.604	**0.611**	0.605	0.607	0.606	0.604	0.604	0.609
	SD	0.017	0.016	0.019	0.019	0.017	0.017	0.016	0.016	0.017	0.018	0.017	0.016	0.017	0.016	0.018	0.018	0.019	0.018
Hepatitis	Average	0.836	0.833	0.832	0.824	0.832	0.828	0.827	0.837	0.828	0.834	0.841	**0.844**	0.828	0.824	0.825	0.836	0.829	0.835
	SD	0.024	0.025	0.025	0.027	0.027	0.028	0.024	0.030	0.027	0.030	0.023	0.024	0.028	0.028	0.025	0.029	0.025	0.025

Table 6
Effect of SCA parameters towards results (p-value).

SCA-PMFWCM	Dataset							
	Heart	Credit	German	Dermatology	Adult	Zoo	Flags	Hepatitis
α	0.009	0.332	0.639	0.676	0.051	0.230	0.354	0.475

Table 7
Effect of GA parameters towards results (p-value).

GA-PMFWCM	Dataset							
	Heart	Credit	German	Dermatology	Adult	Zoo	Flags	Hepatitis
c_r	0.108	0.613	0.961	0.614	0.756	0.444	0.374	0.669
m_r	0.807	0.869	0.229	0.722	0.475	0.108	0.586	0.553
c_r, m_r	0.899	0.818	0.849	0.819	0.995	0.907	0.450	0.961

Table 8
Effect of PSO parameters towards results (p-value).

PSO-PMFWCM	Dataset							
	Heart	Credit	German	Dermatology	Adult	Zoo	Flags	Hepatitis
w	0.768	0.770	0.430	0.051	0.303	0.697	0.936	0.351
c_1	0.973	0.892	0.304	0.873	0.145	0.475	0.826	0.002
c_2	0.643	0.415	0.051	0.836	0.581	0.953	0.616	0.815
w, c_1	0.759	0.160	0.202	0.584	0.207	0.886	0.885	0.351
w, c_2	0.181	0.539	0.983	0.127	0.508	0.776	0.120	0.375
c_1, c_2	0.965	0.110	0.875	0.792	0.469	0.061	0.414	0.994
w, c_1, c_2	0.616	0.198	0.130	0.051	0.921	0.614	0.513	0.277

Table 9
The optimal combinations of tuning parameters.

Method	Tuning parameter	Dataset							
		Heart	Credit	German	Dermatology	Adult	Zoo	Flags	Hepatitis
SCA-PMFWCM	α	2	2	2	2	3	2	2	2
GA-PMFWCM	c_r	0.75	0.75	0.75	0.75	0.9	0.75	0.75	0.75
	m_r	0.05	0.05	0.05	0.05	0.01	0.01	0.01	0.01
PSO-PMFWCM	w	0.8	0.8	0.8	0.8	0.4	0.8	0.8	0.8
	c_1	1	1	1	1	2	0.5	0.5	0.5
	c_2	2	2	2	2	0.5	2	2	2

Table 10
Computational results for numerical datasets.

Dataset		FCM	MFCM	PMFCM	PMFWCM
Iris	*Acc*	0.897	0.901	0.916	**0.923**
	SD	0.000	0.000	0.000	0.000
Wine	*Acc*	0.908	0.923	0.933	**0.935**
	SD	0.005	0.005	0.012	0.013
Haberman	*Acc*	0.512	0.604	0.668	**0.681**
	SD	0.010	0.017	0.031	0.033
Glass	*Acc*	0.708	0.754	0.839	**0.843**
	SD	0.012	0.011	0.015	0.020
Pima Indians	*Acc*	0.644	0.726	0.746	**0.748**
	SD	0.016	0.028	0.031	0.029

results in overall, because the SCA algorithm smoothly transits from exploration to exploitation using an adaptive range in the sine and cosine functions. During the iterative process, the best solution of the global optimum is stored in a variable as the destination point and never get lost during optimization.

6. Case study

6.1. Market segmentation

Market segmentation is the procedure of partitioning a market of potential customers into distinguished groups based on different customers' features. The exploited segments are composed of consumers who will respond similarly to marketing strategies and share similar characteristics such as interests, demands, or careers.

Instead of targeting each potential customer individually, grouping the company's target market into segmented clusters can support the company activity more efficiently in terms of time, money, and others. Partitioning similar customers into one cluster also allows marketers to target specific audiences in a cost-effective manner.

This case study uses the collected mixed attribute dataset of consumers in Thailand. It consists of 353 numbers of data which represents the characteristics and behavior of consumers for soft drinks. The soft drink industry in Thailand affects the direction of food marketing. Therefore, understanding each group of customers will be useful for investing in this industry in the future.

The purpose of this case study is to differentiate the soft drink consumers effectively. The case study only focuses on the products which have the composition of soda and uses a questionnaire to collect the data from consumers. The questionnaire is an online survey and divided into 2 parts, i.e., personal information such as sex, age, education, range of salary, etc., and consumer behavior such as price, advertisement, brand, packaging, ingredient, etc. The questions are set to analyze how consumers think about the soft drink and what kind of soft drink matches them. Table 12 illustrates the data characteristics.

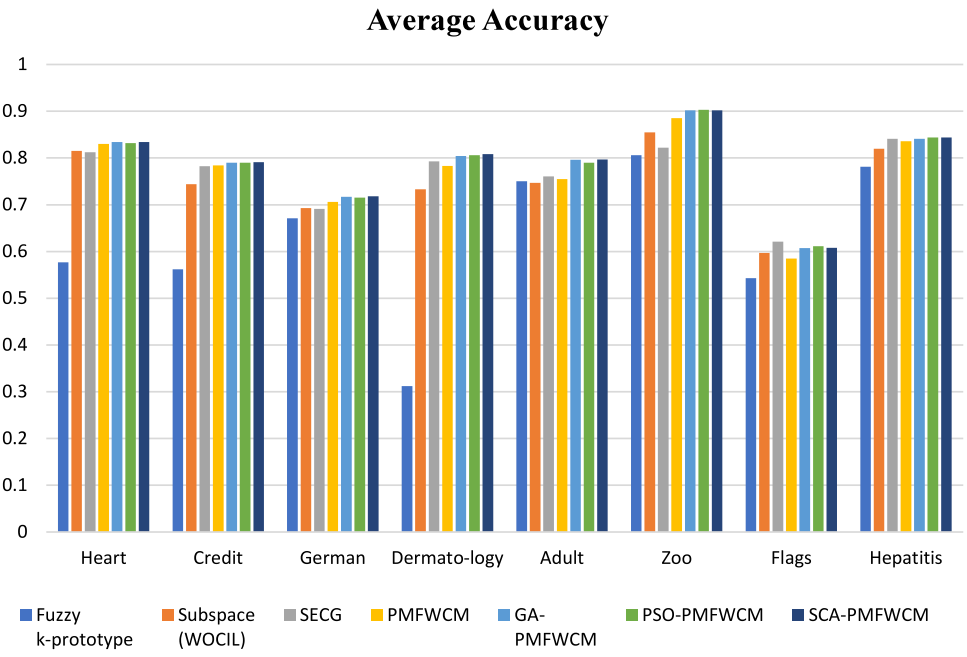

Fig. 2. Comparison of average accuracy among six algorithms on each dataset.

Table 11
Summary of computational results for mixed attribute datasets.

Algorithm / Dataset		Fuzzy k-prototype	WOCIL	SECG	PMFWCM	GA-PMFWCM	PSO-PMFWCM	SCA-PMFWCM
Heart	*Acc*	0.577	0.815	0.813	0.83	**0.834**	0.832	**0.834**
	SD	0	0.052	0.022	0.035	0.017	0.02	0.017
	c.time	1.25	2.45	92.43	21.56	122.23	125.47	120.25
Credit	*Acc*	0.562	0.744	0.783	0.784	0.79	0.79	**0.791**
	SD	0.021	0.062	0.023	0.055	0.033	0.028	0.028
	c.time	1.12	2.22	87.28	20.78	117.71	118.09	116.52
German	*Acc*	0.671	0.693	0.691	0.706	0.717	0.715	**0.718**
	SD	0	0.046	0.037	0.035	0.022	0.028	0.027
	c.time	1.88	3.05	112.24	34.23	165.73	167.12	154.33
Dermatology	*Acc*	0.312	0.733	0.793	0.783	0.804	0.806	**0.808**
	SD	0	0.061	0.028	0.065	0.031	0.031	0.028
	c.time	1.33	2.36	88.33	26.09	125.39	124.45	123.09
Adult	*Acc*	0.75	0.747	0.761	0.755	0.796	0.79	**0.797**
	SD	0.011	0.023	0.015	0.019	0.012	0.019	0.017
	c.time	2.98	5.76	137.36	58.43	243.07	240.22	228.57
Zoo	*Acc*	0.806	0.855	0.822	0.885	0.902	**0.903**	0.902
	SD	0	0.03	0.044	0.027	0.011	0.018	0.013
	c.time	1.01	2.24	86.23	20.34	179.54	190.43	170.45
Flags	*Acc*	0.543	0.597	**0.621**	0.585	0.607	0.611	0.608
	SD	0	0.054	0.028	0.04	0.012	0.016	0.012
	c.time	1.17	2.25	94.98	21.11	195.38	198.82	188.29
Hepatitis	*Acc*	0.781	0.82	0.841	0.836	0.841	**0.844**	**0.844**
	SD	0	0.033	0.019	0.04	0.022	0.024	0.019
	c.time	1.09	2.31	79.22	19.78	163.4	169.97	160.72

Table 12
Soft drink consumer dataset.

	Amount
Number of data	353
Number of attributes	13
Number of numerical attributes	6
Number of categorical attributes	7

6.2. Number of clusters

To decide the number of clusters used in the case study, SSE value is used to predict the number of clusters. SSE is defined as the sum of the squared distance between each member of cluster and its centroid. Fig. 3 used the proposed PMFWCM algorithm to predict the number of clusters. This figure shows that the curve is steepest from cluster 2 to 10. However, the curve becomes smoother from cluster 10 to 11. Thus, the number of clusters is selected at 10. This result will be used as the input parameter for the proposed GA- PMFWCM, PSO- PMFWCM, and SCA- PMFWCM algorithms.

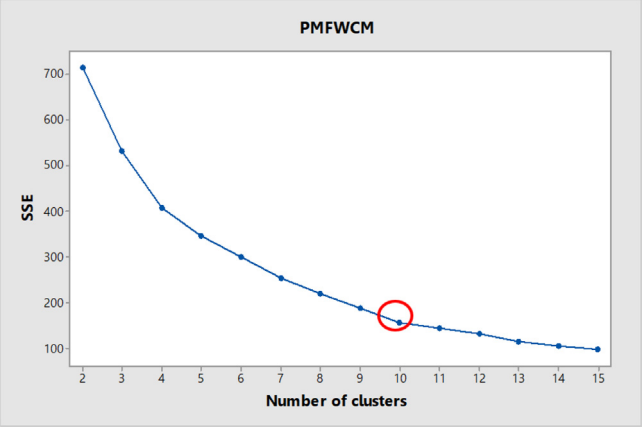

Fig. 3. SSE result of the PMFWCM algorithm in different clusters.

Table 13
SSE results of proposed method.

Methods	PMFWCM	GA-PMFWCM	PSO-PMFWCM	SCA-PMFWCM
Average SSE	156.36	78.96	79.03	77.25
SD	0.102	0.021	0.024	0.018

6.3. Results and discussion

6.3.1. SSE results of proposed methods

Table 13 shows the comparison of SSE results obtained by three proposed methods. Each method was run 30 times. The results shows that SSE values obtained meta-heuristic based PMFWCM algorithms are better than only the one obtained by PMFWCM. It happens because the PMFWCM algorithm sometimes traps in local optima. In contrast, the meta-heuristics can help the PMFWCM avoid local optima and achieve a global solution. The stopping criterion for all algorithms is 100 iterations. From the proposed algorithms, the proposed SCA-PMFWCM gives the smallest SSE. Moreover, the *SD* of SCA-PMFWCM algorithm is more stable.

6.3.2. Clustering results for market segmentation

Since the SCA-PMFWCM is one of the proposed algorithms for this study and gives the good clustering results in verification, the SCA-PMFWCM is used to apply in a real case study for market segmentation.

The consumers are clustered into 10 groups. Based on clustering results, each cluster contains the consumer behavior and characteristics as shown in Table 14.

7. Conclusions

In this study, the PMFWCM algorithm was first proposed for mixed attribute datasets. Thereafter, this study developed the meta-heuristics based PMFWCM algorithm to obtain optimal solutions. There are three meta-heuristic approaches are employed, i.e., SCA, GA, and PSO. These proposed methods can be applied in the real data for market segmentation and consumers could be differentiated and managed effectively.

To evaluate the performance of the proposed methods, the experimental results of this study have been carried out. The clustering algorithms, which combine the PMFWCM algorithm and meta-heuristic approaches, can improve the performance of the PMFWCM algorithm. According to experiments using benchmark and real datasets, the PMFWCM integrated with meta-heuristics is better than the single PMFWCM in terms of *Acc* and SSE. However, the computational time when combining with meta-heuristics longer than without meta-heuristic. Moreover, from the case study results, it can be inferred that the SCA-PMFWCM

Table 14
Characteristics of each cluster.

Cluster #	Consumer behavior and characteristics	Matching product
1	Particularly consider ingredients. The product label is important to them. They also concern about health and nutrient. They usually concentrate and take time to make a decision. They are mostly female and 21–30 years old.	Low calories soft drink, no sugar soft drink
2	The loyal customers. It is quite hard to convince them buying other brands. They are addicted to the same type of products and the same brand. They are mostly female, graduated master's degree, and salary is more than 40,000 baht.	Popular-brand soft drink
3	Combination of Cluster 1 and 2. They are loyal to their favorite brands, but they also consider the ingredients and packaging of the products. They love tasting the new flavor. They are mostly female and graduated high school or bachelor's degree	Fruit-taste soft drink
4	Simple customers. They are quite okay with every kind of products depending on his mood or the situation. They do not have anything to specifically concern. They are mostly 21–30 years old, and graduated bachelor's degree or master's degree.	The simplest kind of soft drink, such as Cola
5	Very low interest in soft drink. They rarely buy it in daily life and mostly are male, graduated bachelor's degree, and >30 years old.	Water, Juice
6	Rarely care about ingredients and price. They do not interest in a commercial advertisement. If they want to buy some products, they will go to decide and compare products by themselves. People in this group are mostly female and salary is more than 35,000 baht.	Unique taste soft drink (Cream soda, Ginger ale)
7	Particularly consider ingredients, yet in terms of tasty feeling. And it must be the flavor that they get used to. People in this group are mostly female, more than 30 years old, and graduated with a bachelor's degree.	Lemon-lime soda, common fruit-taste soft drink
8	Easier to be convinced. They concern about price, ingredients, packaging, and advertisement. Sale promotions always attract them to buy. People in this group are mostly female and graduate master's degree.	Low price soft drink
9	Mostly care about packaging and their preference. If they try once and like it, they will continue buying that product. People in this group are mostly >30 years old and graduated bachelor's degree.	Can or a small bottle of soft drink
10	Usually drink soft drinks in daily life. They drink when they are thirsty or would like to drink together with a meal. People in this group are mostly male and graduated bachelor's degree and salary is less than 15,000 baht.	Cola, Root beer

algorithm gives the smallest SSE and has faster computational time than the GA-PMFWCM and PSO-PMFWCM algorithms

There are several ways to extend this work for future research directions. To evaluate the clustering performance, this study only employs clustering accuracy as the evaluation criteria. In future research, more clustering validation indices should be used such as Adjusted Rank Index (ARI), Normalized Mutual Information (NMI), Davies–Bouldin (DB) index, and so on. Besides, we can consider investigating a parameter-free clustering approach with an unknown number of clusters. Moreover, proposing a new dissimilarity measure for mixed numerical and categorical data is still an attractive issue.

CRediT authorship contribution statement

R.J. Kuo: Supervision, Conceptualization, Methodology. **Patipharn Amornnikun:** Software, Methodology, Data curation, Writing - original draft. **Thi Phuong Quyen Nguyen:** Methodology, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by The University of Danang, University of Science and Technology, code number of Project: T2020-02-16. This support is really appreciated.

References

[1] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson education, Inc, United States of America, 2006.
[2] E. Diday, G. Govaert, Y. Lechevallier, J. Sidi, Clustering in Pattern Recognition, Springer, Netherlands, Dordrecht, 1981, pp. 19–58.
[3] D. Horn, A. Gottlieb, Algorithm for data clustering in pattern recognition problems based on quantum mechanics, Phys. Rev. Lett. 88 (2001) 018702.
[4] Y. Farhang, Face extraction from image based on K-Means clustering algorithms, Int. J. Adv. Comput. Sci. Appl. 8 (2017) 96–107.
[5] Y. Djenouri, A. Belhadi, R. Belkebir, Bees swarm optimization guided by data mining techniques for document information retrieval, Expert Syst. Appl. 94 (2018) 126–136.
[6] T. Nicholls, J. Bright, Understanding news story chains using information retrieval and network clustering techniques, 2018, arXiv preprint arXiv:1801.07988.
[7] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E.D. Trippe, J.B. Gutierrez, K. Kochut, A brief survey of text mining: Classification, clustering and extraction techniques, 2017, arXiv preprint arXiv:1707.02919.
[8] E. Garces, A. Munoz, J. Lopez-Moreno, D. Gutierrez, Intrinsic images by clustering, in: Computer Graphics Forum, Wiley Online Library, 2012, pp. 1415–1424.
[9] G. Arimond, A. Elfessi, A clustering method for categorical data in tourism market segmentation research, J. Travel Res. 39 (2001) 391–397.
[10] H. Tsuen-Ho, C. Kao-Ming, C. Hei-Chun, The fuzzy clustering on market segment, in: Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No.00CH37063), Vol. 622, 2000, pp. 621–626.
[11] C. Morton, J. Anable, J.D. Nelson, Consumer structure in the emerging market for electric vehicles: Identifying market segments using cluster analysis, Int. J. Sustain. Transp. 11 (2017) 443–459.
[12] D. Kamthania, A. Pawa, S.S. Madhavan, Market segmentation analysis and visualization using K-mode clustering algorithm for e-commerce business, J. Comput. Inf. Technol. 26 (2018) 57–68.
[13] K.H. Esbensen, D. Guyot, F. Westad, L.P. Houmoller, Multivariate data analysis: In practice: An introduction to multivariate data analysis and experimental design, in: Multivariate Data Analysis, 2002.
[14] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 1967, pp. 281–297.
[15] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (1984) 191–203.
[16] J. Ji, W. Pang, C. Zhou, X. Han, Z. Wang, A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data, Knowl.-Based Syst. 30 (2012) 129–135.
[17] N.R. Pal, K. Pal, J.M. Keller, J.C. Bezdek, A possibilistic fuzzy c-means clustering algorithm, IEEE Trans. Fuzzy Syst. 13 (2005) 517–530.
[18] B.A. Pimentel, R.M. De Souza, A multivariate fuzzy c-means method, Appl. Soft Comput. 13 (2013) 1592–1607.
[19] B.A. Pimentel, R.M. de Souza, A weighted multivariate fuzzy C-means method in interval-valued scientific production data, Expert Syst. Appl. 41 (2014) 3223–3236.
[20] H. Jia, Y.-M. Cheung, Subspace clustering of categorical and numerical data with an unknown number of clusters, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 3308–3325.
[21] A. Jain, M. Murty, P. Flynn, Data Clustering: A Review ACM Computing Surveys, Vol. 31, Google Scholar, 1999, pp. 264–318.
[22] S.C. Johnson, Hierarchical clustering schemes, Psychometrika 32 (1967) 241–254.
[23] Z. Deng, Y. Jiang, F.-L. Chung, H. Ishibuchi, K.-S. Choi, S. Wang, Transfer prototype-based fuzzy clustering, IEEE Trans. Fuzzy Syst. 24 (2015) 1210–1232.
[24] J. Wang, Z. Deng, K.-S. Choi, Y. Jiang, X. Luo, F.-L. Chung, S. Wang, Distance metric learning for soft subspace clustering in composite kernel space, Pattern Recognit. 52 (2016) 113–134.
[25] Z. Deng, K.-S. Choi, Y. Jiang, J. Wang, S. Wang, A survey on soft subspace clustering, Inf. Sci. 348 (2016) 84–106.
[26] L. Rokach, O. Maimon, Clustering methods, in: Data Mining and Knowledge Discovery Handbook, Springer, 2005, pp. 321–352.
[27] Z. Deng, K.-S. Choi, F.-L. Chung, S. Wang, Enhanced soft subspace clustering integrating within-cluster and between-cluster information, Pattern Recognit. 43 (2010) 767–781.
[28] A. Ahmad, L. Dey, A k-mean clustering algorithm for mixed numeric and categorical data, Data Knowl. Eng. 63 (2007) 503–527.
[29] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Trans. Knowl. Data Eng. 14 (2002) 673–690.
[30] C.-C. Hsu, Y.-P. Huang, Incremental clustering of mixed data based on distance hierarchy, Expert Syst. Appl. 35 (2008) 1177–1185.
[31] S.P. Chatzis, A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional, Expert Syst. Appl. 38 (2011) 8684–8689.
[32] J. Liang, X. Zhao, D. Li, F. Cao, C. Dang, Determining the number of clusters using information entropy for mixed data, Pattern Recognit. 45 (2012) 2251–2265.
[33] K. Zhang, Q. Wang, Z. Chen, I. Marsic, V. Kumar, G. Jiang, J. Zhang, From categorical to numerical: Multiple transitive distance learning and embedding, in: Proceedings of the 2015 SIAM International Conference on Data Mining, SIAM, 2015, pp. 46–54.
[34] W. Chen, Y. Chen, Y. Mao, B. Guo, Density-based logistic regression, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, pp. 140–148.
[35] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Trans. Knowl. Data Eng. (2002) 673–690.
[36] H. Ralambondrainy, A conceptual version of the k-means algorithm, Pattern Recognit. Lett. 16 (1995) 1147–1157.
[37] Z. He, X. Xu, S. Deng, Scalable algorithms for clustering large datasets with mixed type attributes, Int. J. Intell. Syst. 20 (2005) 1077–1089.
[38] A. Ahmad, S. Khan, A survey of state-of-the-art mixed data clustering algorithms, 2019.
[39] J. Fan, M. Han, J. Wang, Single point iterative weighted fuzzy C-means clustering algorithm for remote sensing image segmentation, Pattern Recognit. 42 (2009) 2527–2540.
[40] W. Pedrycz, P. Rai, Collaborative clustering with the use of fuzzy C-means and its quantification, Fuzzy Sets Syst. 159 (2008) 2399–2427.
[41] L. Himmelspach, S. Conrad, A Possibilistic Multivariate Fuzzy C-Means Clustering Algorithm, Springer International Publishing, Cham, 2016, pp. 338–344.
[42] F. Glover, Future paths for integer programming and links to artificial intelligence, Comput. Oper. Res. 13 (1986) 533–549.
[43] I.H. Osman, J.P. Kelly, Meta-heuristics: An overview, in: Meta-Heuristics, Springer, 1996, pp. 1–21.
[44] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT press, 1992.
[45] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, 1988.
[46] J.C. Bezdek, S. Boggavarapu, L.O. Hall, A. Bensaid, Genetic algorithm guided clustering, in: Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, IEEE, 1994, pp. 34–39.
[47] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognit. 33 (2000) 1455–1465.

[48] C.A. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms, 1996.
[49] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Micro Machine and Human Science, 1995. MHS'95. Proceedings of the Sixth International Symposium on, IEEE, 1995, pp. 39–43.
[50] H.T. Dorrah, A.M. El-Garhy, M.E. El-Shimy, PSO based optimized fuzzy controllers for decoupled highly interacted distillation process, Ain Shams Eng. J. 3 (2012) 251–266.
[51] S. Mirjalili, SCA: A sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[52] E. Michielssen, S. Ranjithan, R. Mittra, Optimal multilayer filter design using real coded genetic algorithms, IEE Proc. J 139 (1992) 413–420.
[53] X. Zhao, F. Cao, J. Liang, A sequential ensemble clusterings generation algorithm for mixed data, Appl. Math. Comput. 335 (2018) 264–277.