Expert Systems With Applications xxx (xxxx) xxx

Contents lists available at ScienceDirect

Expert Systems With Applications

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

initKmix-A novel initial partition generation algorithm for clustering mixed data using k-means-based clustering

Amir Ahmad [a,*], Shehroz S. Khan [b]

[a] College of Information Technology, United Arab Emirates University, Al-Ain, United Arab Emirates
[b] KITE, University Health Network, 550, University Avenue, Toronto, ON, Canada

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Mixed data
Clustering
k-means
Initialization
Random

ABSTRACT

Mixed datasets consist of both numeric and categorical attributes. Various k-means-based clustering algorithms have been developed for these datasets. Generally, these algorithms use random partition as a starting point, which tends to produce different clustering results for different runs. In this paper, we propose, *initKmix*, a novel algorithm for finding an initial partition for k-means-based clustering algorithms for mixed datasets. In the *initKmix* algorithm, a k-means-based clustering algorithm is run many times, and in each run, one of the attributes is used to create initial clusters for that run. The clustering results of various runs are combined to produce the initial partition. This initial partition is then used as a seed to a k-means-based clustering algorithm to cluster mixed data. Experiments with various categorical and mixed datasets showed that *initKmix* produced accurate and consistent results, and outperformed the random initial partition method and other state-of-the-art initialization methods. Experiments also showed that k-means-based clustering for mixed datasets with *initKmix* performed similar to or better than many state-of-the-art clustering algorithms for categorical and mixed datasets.

1. Introduction

Clustering is a process in which similar data points are grouped in the same clusters whereas dissimilar data points are grouped is different clusters based on some notion of 'similarity' (Jain & Dubes, 1988). Many datasets only contain numeric or categorical attributes; however, the majority of real-world datasets contain both types of attributes. These are called mixed datasets (Bishop, 2008; Witten & Frank, 2005). Most clustering algorithms developed to handle only numeric or categorical datasets cannot be directly used to cluster mixed datasets because the calculation of 'similarity' is not straight forward in mixed datasets (Ahmad & Khan, 2019).

Various types of clustering algorithms have been developed to handle mixed datasets (Balaji & Lavanya, 2018); the most prominent being the partitional, hierarchical, model-based, and neural network-based methods (Ahmad & Khan, 2019). The partitional clustering methods are more popular among the research community because they are (i) simpler in interpretation and implementation, (ii) linear in time complexity with the number of data points, and (iii) easily adaptable to parallel architectures. The traditional k-means algorithm is a partitional clustering algorithm that was developed to cluster datasets with only numeric attributes (Khan & Ahmad, 2004). Its objective function uses a distance metric (e.g. Euclidean distance) that can only be defined for numeric attributes. This type of clustering algorithm is further extended to develop **k**-means-based clustering for **m**ixed **d**atasets (KMD) algorithms (Ahmad & Dey, 2007; Ahmad & Khan, 2019; Huang, 1997a). The KMD algorithms comprise of a family of algorithms that may differ in the definition of their cluster centre, distance measure and objective function (Ahmad & Khan, 2019). Most of these partitional clustering approaches perform hard clustering, i.e. a data point can belong to only one cluster. In fuzzy clustering approaches, a data point can be assigned to more than one cluster with different membership values. Approaches based on fuzzy clustering have also been applied for mixed datasets (D'Urso & Massari, 2019; Ji et al., 2012). In this paper, we focus on hard clustering algorithms.

In general, there are two type of approaches for initializing the k-means clustering algorithm. In the first approach, the random initial cluster centres, k (the number of clusters) data points are selected randomly, which act as the initial cluster centres. In the second approach, the random initial partition, first randomly assigns a cluster to each data point and then compute the centres of these clusters. The k-means based clustering algorithms may suffer from several drawbacks (Khan & Ahmad, 2004). Prominently, the k-means optimization function can stuck in the local minima; therefore, with a different

[*] Corresponding author.
E-mail addresses: amirahmad@uaeu.ac.ae (A. Ahmad), shehroz.khan@uhn.ca (S.S. Khan).

https://doi.org/10.1016/j.eswa.2020.114149
Received 2 December 2019; Received in revised form 2 September 2020; Accepted 21 October 2020
Available online 27 October 2020

initialization, the k-means clustering algorithm may lead to different final clusters. Hence, it is difficult to obtain reliable and consistent clustering results (Khan & Ahmad, 2004).

Similar to the standard k-means clustering algorithm for numeric data, KMD algorithms also suffers from the random initialization problem. A few initialization methods have been developed for KMD algorithms (Ahmad & Hashmi, 2016; Chen et al., 2017; Ji et al., 2015, 2015a; Wangchamhan et al., 2017). However, these methods are either computationally expensive (quadratic complexity with respect to the number of data points) (Chen et al., 2017; Ji et al., 2015, 2015a) or do not produce consistent clustering results (Ahmad & Hashmi, 2016; Wangchamhan et al., 2017). In this paper, our baseline KMD algorithm is the one proposed by Ahmad and Dey (2007), **k**-**m**eans **c**lustring for **m**ixed **d**atasets with a mixed distance measure (KMCMD). The reason to choose KMCMD is that it has shown superior performance in comparison to other similar partitional clustering algorithms to cluster mixed datasets (Ahmad & Dey, 2007).

In this paper, we present the *initKmix* algorithm, a novel algorithm to compute the initial partition for KMCMD algorithm. The initial partition is then fed to KMCMD algorithm to cluster mixed datasets. The initial clusters produced by the *initKmix* algorithm are stable across various runs of the algorithm or when the order of the data points is changed. This ensures consistent and reliable clustering results. Moreover, the time complexity of the *initKmix* algorithm is linear w.r.t. the number of data points; thus, it can be used with large datasets. The *initKmix* algorithm does not guarantee to find the global optima of the optimization functions of the KMCMD algorithm; however, the experiments suggest that the KMCMD algorithm with the *initKmix* algorithm produces superior clustering results. The choice of k is an important issue in the KMCMD algorithm (Liang et al., 2012); however, in this paper, we focus on finding the appropriate initial partitions with given value of k.

The paper is organized in the following way. Section 2 presents related work focusing on the methods to calculate the initial partition for KMCMD algorithms for mixed datasets. The *initKmix* initial partition method is presented in Section 3. Results are presented in Section 4, followed by conclusions and future research directions in Section 5.

2. Related work

The k-means clustering algorithm is a commonly used clustering algorithm for datasets consisting of numeric attributes because of its low computational complexity (MacQueen, 1967). Its complexity is linear with respect to the number of data points and scales well for large datasets. The algorithm minimizes the following optimization function (Eq. (1)) iteratively,

$$\sum_{i=1}^{n} \xi(d_i, C_i) \tag{1}$$

where n is the number of data points in the dataset, C_i is the nearest cluster centre to data point d_i, ξ is a chosen distance measure (or similarity measure) between d_i and C_i. Generally, the Euclidean distance is used as the distance measure.

k-means clustering algorithm computes cluster centres and data point memberships at each iteration. The algorithm starts with user-defined initial clusters. Generally, a random initial partition is selected that may produce different clustering results for different runs of the algorithm. To overcome this problem various methods have been proposed for the computation of the initial partition (Arthur & Vassilvitskii, 2007; Bradley & Fayyad, 1998; Duda et al., 1973; Khan & Ahmad, 2004). The k-means clustering algorithm can only handle pure numeric datasets. For pure categorical datasets, k-modes clustering algorithm is proposed (Huang, 1997b), in which the cluster centre is represented by the mode of the attribute values of the data points presented in that cluster and the Hamming distance is used to compute the distance between a data point and a cluster centre. Similar to the k-means clustering algorithm, k-modes clustering algorithm can also show inconsistent clustering results due to the choice of random initial partition. Various algorithms have been proposed to find the appropriate initial partition for K-modes clustering algorithm. (He, 2006; Khan & Ahmad, 2003, 2013; Khan & Kant, 2007; Wu et al., 2007).

Algorithm 1 General steps of a KMD algorithm with the random partition.

```
Input- Mixed dataset $T$, the number of data points is $n$, the number of attributes is $m$, the number of clusters is $k$.
Begin
1- Assign all data points to $k$ clusters randomly
2- Repeat steps $A$ and $B$
(A) Calculate the centres of the clusters.
(B) Each data point is assigned to its nearest cluster using the user defined distance measure.
Until data points no longer change cluster membership or a predefined number of iterations is reached.
End
```

Several KMD algorithms have been developed to extend the k-means clustering algorithm to mixed datasets (Ahmad & Dey, 2007; Huang, 1997a). In these algorithms, new definitions for cluster centres and distances are proposed to handle mixed datasets (Ahmad & Dey, 2007; Huang, 1997a). The general steps in these algorithms algorithm are presented in Algorithm 1. A comprehensive review of these algorithms can be found in the survey paper of Ahmad and Khan (Ahmad & Khan, 2019). Due to their similarity with k-means clustering, these algorithms also suffer from the issues of random initialization and finding appropriate number of clusters (k). The focus of this paper is to find initial partition for a KMD algorithm (the KMCMD algorithm (Ahmad & Dey, 2007)). Therefore, in this section, we restrict our literature survey to the research works that attempted to initialize KMD algorithms.

Ji et al. (2015) suggest a method to determine initial clusters for a KMD algorithm by computing compute density and distances between data points. In another work, Ji et al. (2015a) propose the concept of centrality based on neighbouring points and combined it with the distances between data points to compute initial clusters. However, these algorithms have quadratic complexity and are therefore may not work efficiently for large mixed datasets. Using density peaks (Rodriguez & Laio, 2014), Chen et al. (2017) propose an algorithm to determine the initial cluster centres for mixed datasets. Higher density points are used to identify cluster centres. This algorithm also has quadratic complexity. Wangchamhan et al. (2017) combine a search algorithm, the League Championship algorithm (Kashan, 2009), with a KMD algorithm to identify the initial cluster centres. This algorithm has many parameters; hence, the final clustering results are dependent on parameter settings, where different parameters may lead to different clustering results. Zheng et al. (2010) combine the evolutionary algorithm (EA) with the k-prototypes clustering algorithm (Huang, 1998). The global searching capability of EA leads the proposed algorithm to be stable to cluster initialization. However, clustering results are dependent upon the parameters and not consistent in different runs.

The k-Harmonic means clustering algorithm addresses the random initial clusters problem applying a cost function (Zhang, 2001) for numeric datasets which uses the harmonic means of the distances from each data point to the centres. This algorithm creates clusters that are more stable than those generated by k-means clustering algorithm with random initial clusters. Ahmad and Hashmi (2016) combine the distance measure and the definition of cluster centres for mixed datasets suggested by Ahmad and Dey (2007) with the k-Harmonic clustering algorithm (Zhang, 2001) to develop the k-Harmonic clustering algorithm for mixed datasets. Their method is less sensitive to the choice of the initial cluster centres. The standard deviation of the clustering accuracy of this method is small in comparison to that of the random initialization method. However, clustering results are not stable for runs with different initial partitions.

The literature review suggests that the existing initialization methods for KMD algorithms are either computationally expensive or do not produce consistent clustering results. This further limits the use of existing methods in real world situations where an algorithm's execution time and reliability of clustering results are the key factors for their adoption. In the next section, we present our proposed algorithm *initKmix* to compute the initial partition for a KMD algorithm (KMCMD algorithm Ahmad & Dey, 2007).

3. *Initkmix* algorithm

The *initKmix* algorithm is based on following two experimental observations noted in previous research works in k-means clustering (Khan & Ahmad, 2004) and k-modes clustering (Khan & Ahmad, 2013):

(i) Some data points in a given dataset have similar final cluster membership irrespective of the initial partition (Khan & Ahmad, 2004, 2013). This observation has been used to determine the initial partition for k-means clustering algorithm (for pure numeric datasets) (Khan & Ahmad, 2004) and k-modes clustering algorithm (for pure categorical datasets) (Khan & Ahmad, 2013). We extend this approach to determine the initial partition for the KMCMD algorithm.

(ii) Each of the attributes of a dataset can contribute to final clustering result. Therefore, an individual attribute may be used to determine the initial partition (Khan & Ahmad, 2004, 2013).

The *initKmix* algorithm generates multiple instances of clustering, this aspect is similar to multiple-view clustering (Muller et al., 2012; Yang & Wang, 2018). Multiple-view clustering manages the production of different clustering results for datasets generated from different sources or observed from different views. These clustering results are combined to generate a clustering result. The diversity of the various views is an important aspect of this clustering approach. It is suggested that multiple views should be used to present data points more comprehensively and accurately (Muller et al., 2012; Yang & Wang, 2018). *initKmix* algorithm uses a similar approach as it creates multiple clustering and for each clustering a different view is used to create the initial partition.

The *initKmix* algorithm has two important components, running KMCMD algorithm m times (m is the number of attributes) to generate m instances of clustering. Each instance of clustering creates a cluster label for each data point. m instances of clustering generate a string of m cluster labels for each data point (Table 1). These m instances of clustering (strings of m cluster labels) are combined to generate a clustering, which is then used as initial partition for the KMCMD algorithm. As discussed earlier, a mixed dataset contains two types of attributes; categorical and numeric. We use each of these m attributes to create the initial partition in one of the runs of the KMCMD algorithm. These m runs of the KMCMD algorithm generate m clustering results. The clustering ensemble algorithm (Strehl & Ghosh, 2003) is then used to combine these m results to produce final clustering that is used as the initial partition for the KMCMD algorithm. In one run, one attribute is used to create the initial partition that can be considered as one view of the data. Multiple-views of the data by using different initial partitions are used to create multiple instances of clustering. These views are diverse as they use different attributes to create the initial partition. Hence, we expect that combining these different clustering results will generate an accurate clustering that can be used as the initial partition for the KMCMD algorithm. The specific steps are presented in Algorithm 2. We will discuss each step of the proposed method in detail below.

Algorithm 2 The proposed *initKmix* algorithm for creating initial partition for the KMCMD algorithm.

```
Input- Mixed dataset $T$, the number of attributes is $m$, the number of numeric attributes is $m_r$ and the number of categorical attributes is $m_c$ ($m = m_r + m_c$), $k$ is the number of desired clusters.
Begin
1(a)- For numeric attributes
for i=1...$m_r$ do
    Create a clustering result by using $KMCMD\_Initial\_Numeric$ algorithm (Algorithm 3).
end for
1(b)- For categorical attributes
for i=1...$m_c$ do
    Create a clustering result by using $KMCMD\_Initial\_Categorical$ algorithm (Algorithm 4).
end for
/*We will have $m$ clustering results as presented in Table 1 */
2- Combine these $m$ clustering results by using a clustering ensemble algorithm discussed in Section 3.3 to get $k$ clusters.
/*These $k$ clusters will be treated as the initial partition for the KMCMD algorithm.*/
End
```

Algorithm 3 $KMCMD\_Initial\_Numeric$ algorithm for creating clusters by using the KMCMD algorithm with the initial partition created using a numeric attribute.

```
Input- Mixed dataset $T$, the KMCMD algorithm, $k$ is the number of desired clusters.
Begin
For $i$th numeric attribute
/*Assume that the $i$th attribute is normally distributed*/
1- Convert it to the standard normal distribution using the following steps
(i)- Calculate the mean $\mu$ for the $i$th numeric attribute
(ii)- Calculate the standard variation $\sigma$ for the $i$th numeric attribute.

(iii)- Calculate the z-score of each value ($x$) of the $i$th attribute using the formula $\frac{x-\mu}{\sigma}$.
2- Find out $k$-1 values of z for which the area between each range is equal to $\frac{1}{k}$. We use -∞ and +∞ as the boundaries of extreme ranges.

3- Divide the dataset $T$ into $k$ clusters, depending on the range in which the attribute value falls.
4- Use these clusters as the initial partition for the KMCMD algorithm and cluster the mixed data $T$ with the KMCMD algorithm.
End
```

3.1. *Initial partition using numeric attributes*

Each numeric attribute is used to create initial clusters for one of the runs of the KMCMD algorithm. A numeric attribute is assumed to have a normal distribution (Khan & Ahmad, 2004). Initial clusters are created such that the probability distributions of the attribute values are equal in each cluster. For k clusters, k-1 boundaries in the normal distribution graph are created so that the area between two adjacent boundaries is $\frac{1}{k}$. The extreme points $-\infty$ and $+\infty$ will also be used as boundaries along with the $k-1$ boundaries. For example, if we want to create three clusters from a numeric attribute, two points z_1 and z_2 are selected such that the area between $-\infty$ and z_1, z_1 and z_2, and z_2 and $+\infty$ is $\frac{1}{3}$. The data points are distributed in three clusters depending on the range in which the attribute value of a data point lies to create initial partition. Our proposed method is different from the method proposed by Khan and Ahmad (2004) to compute the initial partition for k-means clustering algorithm as the latter computes the initial cluster centres by

Algorithm 4 *$KMCMD\_Initial\_Categorical$* algorithm for creating clusters by using the KMCMD algorithm with the initial partition created by using a categorical attribute.

```
Input- Mixed dataset $T$, the KMCMD algorithm.
Begin
For $i$th categorical attribute
1- Find unique attribute values,
2- Create clusters based on attribute values such that data points
   having the same attribute value for the $i$th categorical attribute will
   be in the same cluster.
3- Use the output of the last step as the initial partition for the
   KMCMD algorithm and cluster the mixed dataset $T$.
/*As the number of categorical values may be different from the
value of $k$. The final number of clusters may be different than the
value of $k$*/
End
```

selecting a point in a given range such that the area under the curve in that range is divided equally. These centres are used to create the initial clusters for a run of the k-means clustering algorithm. However, as the normal distribution curve is not a straight line parallel to the horizontal axis, the probability distributions of the attribute values in the clusters are not equal. However, in the proposed method, the boundaries are computed in a way that the probability distributions of attribute values in the clusters are equal. Here, we would like to point out that we assume that all the numeric attributes are normally distributed. Previous results suggest that the assumption of numeric attributes to follow normal distribution works well, in practice, in finding initial partitions by k-means clustering (Khan & Ahmad, 2003, 2004).

The KMCMD algorithm is run on the complete mixed dataset with the initial partition created by the numeric attribute resulting in a clustering result. The algorithm for creating clusters by using the KMCMD algorithm with the initial partition yielded using a numeric attribute is presented in Algorithm 3.

3.2. *Initial partition using categorical attributes*

A categorical attribute consists of two or more categorical values. It has been shown that these attribute values can be used to create clusters (He et al., 2005; Iam-On et al., 2012; Khan & Ahmad, 2013). Following a similar methodology, we use the values of a categorical attribute to create an initial partition. For example, for an attribute with three attribute values α, β and γ, the points can be clustered in the three clusters based on these attribute values; these clusters are then used as initial clusters. Each categorical attribute is used to create the initial clusters for one of the runs. Khan and Ahmad (2013) use a similar approach to identify the initial partition for pure categorical datasets. They generally only use attributes with values that are equal to or less than the number of desired clusters to prevent a large number of distinct clustering strings from being created (a clustering string is a combination of all cluster labels for a data point, see Table 1). In contrast, the proposed approach has no such constraint; all the categorical attributes all used to create the initial partition. The *initKmix* algorithm is for mixed datasets with two kinds of attributes, numeric and categorical. It is important that both types of attributes are treated equally. As we use each numeric attribute to create the initial partition in one of the runs of the KMCMD algorithm for mixed datasets, each categorical attribute should also be used in one of the runs of the KMCMD algorithm, which is not the case in the approach proposed by Khan and Ahmad (2013).

The KMCMD algorithm is run on the complete mixed dataset with the initial partition created by the categorical attribute to yield a clustering result. The method for creating clusters using the KMCMD algorithm for mixed datasets with initial partition created using a categorical attribute is presented in Algorithm 4.

Table 1
An example of clustering results after step 1 of Algorithm 2. For a dataset with four attributes and five data points. A column represents a clustering result in a run. Four cluster labels will be generated for each data point. For example, in the first, second, third and fourth run, the first data point is given a cluster label a, b, b, and a respectively.

Data point	First run	Second run	Third run	Fourth run
1	a	b	b	a
2	b	a	b	a
3	b	a	b	a
4	b	b	a	b
5	a	b	a	b

3.3. *Combining multiple clustering results*

In the *initKmix* algorithm, the KMCMD algorithm is run m times to produce m clustering results. An example of clustering results for different runs is presented in Table 1. These m clustering results are combined to yield the initial partition. A similar approach has been used by Khan and Ahmad (2004, 2013), however, their method to combine m clustering results can be quadratic with respect to the number of data points in the worst case.

Several cluster ensemble algorithms have been developed to combine the results of multiple clustering results of a given dataset (Fig. 1), resulting in superior clustering results (Ghosh & Acharya, 2011; Strehl & Ghosh, 2003; Topchy et al., 2005). Strehl and Ghosh (Strehl & Ghosh, 2003) propose cluster ensemble algorithms that have a linear time complexity to the number of data points. We use the following cluster ensemble algorithms to combine the m clustering results to obtain the initial clusters.

(i) HyperGraph partitioning algorithm (Strehl & Ghosh, 2003) - In this algorithm, the clustering ensemble problem is defined as the partitioning problem of a hypergraph, where hyperedges (a hyperedge is a generalization of an edge that can connect any set of vertices) represents clusters. The complexity of this method is $O(nkr)$ where r is the number of runs of the clustering algorithm.

(ii) Meta-CLustering algorithm (Strehl & Ghosh, 2003) - In this algorithm, the cluster ensemble problem is considered to be the cluster correspondence problem. Groups of similar clusters are identified and combined. The complexity of this method is $O(nk^2r^2)$.

3.4. *The KMCMD algorithm*

As mentioned in Section 1, the k-means algorithm cannot directly be used to cluster mixed data because of the distance function in the objective function it optimizes. Ahmad and Dey (2007) propose KMCMD that modifies the distance function of the standard k-means algorithm. In this paper, we use this algorithm in the *initKmix* algorithms as a baseline to determine the initial partition. This initial partition will then be used with the KMCMD algorithm to produce the final clustering.

In this algorithm, Ahmad and Dey (2007) propose a distance measure for categorical attribute values. The weight of a numeric attribute, which represents the significance of the attribute, is also incorporated in the distance function to highlights its significance. A novel frequency-based definition of the cluster centre for categorical attributes is also proposed for a better representation of clusters. The modified distance function (Ahmad & Dey, 2007) computing the distance ($\psi(d_i, C_j)$) between the ith data point (d_i) and the jth cluster centre (C_j) is given as

$$\psi(d_i, C_j) = \underbrace{\sum_{t=1}^{m_r}(w_t^r(d_{it}^r - C_{jt}^r))^2}_{numeric} + \underbrace{\sum_{t=1}^{m_c}(\Omega(d_{it}^c, C_{jt}^c))^2}_{categorical} \qquad (2)$$

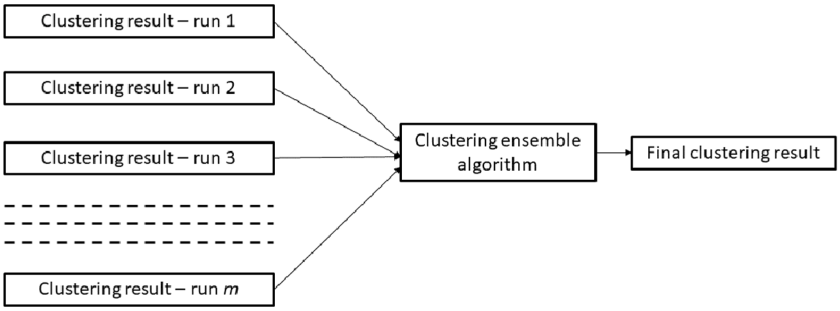

Clustering result – run 1
Clustering result – run 2
Clustering result – run 3
Clustering result – run m
Clustering ensemble algorithm
Final clustering result

Fig. 1. Multiple m clustering results are combined to get one clustering result, which is used as initial clusters..

where w_t is the weight of the tth numeric attribute and d^r_{it} is the value of the tth numeric attribute of the ith data point. C^r_{jt} is the value of the tth numeric attribute of the jth cluster centre. d^c_{it} is the value of the tth categorical attribute of the ith data point. C^c_{jt} is the centre representation of the jth centre for the tth categorical attribute. Ω is the distance between a cluster centre and a data point for a categorical attribute. As shown in Eq. (2), there are two terms, one each for computing the distance for the numeric and categorical attributes. For numeric attributes, the Euclidean distance with the weight of each numeric attribute is used. For categorical attributes, the frequency based definition for centre and co-occurrence based method to compute the distance between two attribute values is used.

This distance measure does not take the distance between two attribute values of a categorical attribute as 0 or 1 (Hamming distance), rather it computes the distance between two values of an attribute from the dataset. The distance between two attribute values α and β with respect to the other attribute is computed by using the following formula

$$max \mid p(f|\alpha) + p(\sim f|\beta) \mid -1 \qquad (3)$$

where $p(f|\alpha)$ represents the probability for data point with attribute value α having other attribute values belonging to a subset of classes f, whereas $p(\sim f|\beta)$ represents the probability for data points with attribute value β having other attribute values not belonging to f. Out of many subsets of classes, a subset maximizing the value in Eq. (3) is selected. The distance between two values of an attribute is computed with respect to all the other attributes and the average is taken as the distance between these two attribute values.

The distance algorithm does not take the significance of the numeric attributes as equal but computes the significance of a numeric attribute from the dataset. A numeric attribute is discretized; the new attribute is treated as a categorical attribute. The average of the distances of all the pairs of attribute values is taken as the weight of the numeric attribute. The discretization of numeric attributes is undertaken only to compute the weight of the numeric attributes. The clustering is performed with numeric attributes. The complete clustering algorithm is presented in Algorithm 5.

Algorithm 5 The KMCMD algorithm (Ahmad & Dey, 2007).

```
Input- Mixed dataset $T$, the number of attributes $m$, the number of numeric attributes $m_r$ and the number of categorical attributes $m_c$. $k$ the number of desired clusters.
Begin
For all numeric attributes
for i=1...$m_r$ do
    1- Normalize all numeric attributes.
    2- Discretize all numeric attributes.
    /*It will be considered as categorical attribute.*/
end for
For all categorical attributes (categorical attributes in the original dataset and discretized numeric attributes)
for i=1...$m$ do
    1- Compute the distance between every pair of attribute values by using the co-occurrence of the pair with respect to other attributes (using Eq. (3)).
end for
For all numeric attributes
for i=1...$m_r$ do
    1- Compute the distance between every pair of discretized attribute values.
    2- The average of all the distances (between each pair of attribute values) is taken as the weight of the numeric attribute.
end for
/*Following steps are for clustering*/
(1) Take the original dataset that has normalized numeric attributes and categorical attributes, assign the data points to $k$ clusters randomly.
(2) Repeat steps $A$ and $B$
(A) Calculate the centres of the clusters.
/*For a numeric attribute, the mean is used to define a cluster centre whereas for a categorical attribute. A frequency-based measure is used to define a cluster centre.*/
(B) Each data point is assigned to its nearest cluster using the distance measure defined in Eq. (2).
Until no data point changes cluster membership or the predefined number of iterations is reached.
End
```

3.5. Computational complexity

The *initKmeans* algorithm run the KMCMD algorithm (Ahmad & Dey, 2007) m times for a dataset (with n data points) to create m clustering results corresponding to each m attributes. The complexity of the KMCMD algorithm is $O(m^2n + m^2S^3 + pn(km_r + km_cS))$ where p is the number of iterations and S is the average number of distinct categorical values. Hence, for m number of runs, the complexity is $O(m(m^2n + m^2S^3 + pn(km_r + km_cS)))$. Then these m clustering results are combined to obtain a clustering result. These results are combined using HyperGraph partitioning algorithm (complexity $O(nkm)$) or Meta-CLustering algorithm (complexity $O(nk^2m^2)$) (Strehl & Ghosh, 2003). This clustering result is used as an initial partition to run the KMCMD algorithm. Hence, the total complexity when clustering results are combined with:

HyperGraph partitioning algorithm is

$$O(((m+1)(m^2n + m^2S^3 + pn(km_r + km_cS))) + O(nkm))$$
$$\implies O((m(m^2n + m^2S^3 + pn(km_r + km_cS))) + (nkm))$$

Meta-CLustering algorithm is

$$O(m(m^2n + m^2S^3 + pn(km_r + km_cS)) + O(nk^2m^2))$$

In both the cases, the time complexity is linear to the number of data points. In the experiment, both ensemble algorithms are run and the clustering results with better normalized mutual information is selected. As both the ensemble algorithms are linear to the number of data points, the total complexity of the clustering algorithm will remain linear to the number of data points.

4. Experiments and results

We implemented the *initKmix* and the KMCMD algorithm (Ahmad & Dey, 2007) using Java JDK 1.8. To perform cluster ensemble step, the Octave implementations of the cluster ensemble algorithms were used (Strehl, 2011). A minor modification was made to the clustering ensemble implementation such that the method only considered Hyper-Graph Partitioning Algorithm and Meta-CLustering Algorithm based on the maximum average normalized mutual information (Strehl & Ghosh, 2003). The *initKmix* algorithm was first tested on a simulated mixed dataset, then on four pure categorical datasets and five mixed datasets downloaded from UCI repository (Dua & Graff, 2017). All these datasets have predefined class and class labels, which were taken as ground truth. The number of the desired clusters was set to the number of the classes. The clustering accuracy was computed against the ground truth. In the experiments, the performance of a clustering algorithm was measured in terms of clustering quality. Three different clustering performance measures (clustering accuracy, Rand index and adjusted Rand index) were used for that purpose.

Each cluster was mapped to a distinct class so that the following measure (Ahmad & Dey, 2007) had the maximum value;

$$AC = \frac{\sum_{i=1}^{K} G_i}{n} \tag{4}$$

Where G_i is the number of data points correctly assigned to a class.

This measure is called clustering accuracy (AC) and has been used to compare the clustering results (Ahmad & Dey, 2007). Two measures derived from AC, $\overline{AC}$ (for the average clustering accuracy) and SD (for the standard deviation of clustering accuracy), were also used to present the results of clustering methods with random initial clusters; The average clustering accuracy for T runs is defined in the following way;

$$\overline{AC} = \frac{\sum_{i=1}^{T} AC_i}{T} \tag{5}$$

where AC_i is the clustering accuracy in the ith run.

The standard deviation of the clustering accuracy for T runs is computed in the following way;

$$SD_{AC} = \sqrt{\frac{\sum_{i=1}^{T} (AC_i - \overline{AC})^2}{T}} \tag{6}$$

Where AC_i is the clustering accuracy in the ith run.

Two other clustering performance measures, Rand index (RI) and adjusted Rand index (ARI) (Rand, 1971) were also employed to compute the clustering performance. RI represents the frequency of occurrence of agreements between two clustering over the total pairs of data points. RI is defined by following expression;

$$RI = \frac{a+b}{\binom{n}{2}} \tag{7}$$

where a in the number of pairs of data points belong to the same cluster across two different clustering results and the b is the number of pairs of data points are in different clusters across two different clustering results. Classes were taken as one clustering and the clustering is compared with it.

ARI is the corrected-for-chance version of the Rand index. ARI is defined by following expression;

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{8}$$

n_{ij} denotes the number of data points common between ith cluster of clustering and jth class, a_i refers the number of data points in ith cluster and b_i denotes the number of data points in jth class.

Similar to $\overline{AC}$, for many runs of the KMCMD algorithm, the average values of RI ($\overline{RI}$) and ARI ($\overline{ARI}$) were computed. Standard deviation of RI (SD_{RI}) and ARI (SD_{ARI}) were also used to compute the performance of clustering algorithms.

The higher values of performance measures (AC, RI and ARI) suggest better clustering results. The maximum possible value for these performance measures is 1 that represents that the data clustering and data classes are exactly the same. Similarly, for the KMCMD algorithm with random initial partitions, high values of ($\overline{AC}$, $\overline{RI}$ and $\overline{ARI}$) are desired. SD represents the inconsistency of clustering results with a different initialization in each run of the KMCMD algorithm. The low values of SD (SD_{AC}, SD_{RI} and SD_{ARI}) point to the robustness of the algorithm for different initial partitions.

We carried out the two types of experiments. First, we compared the *initKmix* algorithm against the random initial partition method. The KMCMD algorithm (Ahmad & Dey, 2007) was run 50 times with a random initial partition method and average clustering performances are presented. We ran the *initKmix* algorithm just once for a dataset to obtain the initial partition, and then the KMCMD algorithm (Ahmad & Dey, 2007) was applied with this initial partition to yield the clustering result. As this approach yields identical results every time, the SD of clustering results for the *initKmix* algorithm is 0 for any dataset.

Second, the KMCMD algorithm (Ahmad & Dey, 2007) with *initKmix* algorithm was also compared with other clustering algorithms using AC performance measure. Results for these clustering algorithms were taken from the published papers. Majority of papers on mixed data clustering methods use similar datasets and performance measure (AC). Therefore, we also use those frequently used datasets and the performance measure to facilitate comparison of various clustering algorithms with the KMCMD algorithm with the *initKmix* algorithm.

Now, we show the results on a mixed simulated dataset, followed by results on pure categorical and mixed datasets.

4.1. Simulated mixed data

We compared the performance of *initKmix* algorithm against random initial partition on a simulated mixed dataset. Following *clustMixType* package (Szepannek & Aschenbruck, 2020) of R (R Core Team, 2013), a mixed dataset was generated. The dataset had four attributes, two of them were categorical, whereas the other two were numeric. Each categorical attribute had two categories, A and B for the first categorical attribute and X and Y for the second attribute. The numeric attributes were created using normal distribution. There were 400 data points divided equally into four clusters. First attribute values of cluster, C_1, were created by selecting attribute categories A and B randomly with 0.9 and 0.1 probability respectively. Similarly the second attribute values of C_1, were created by selecting attribute categories X and Y. Third attribute values were created using normal distribution with μ = -5 and σ = 1. The fourth attribute values generated using the same method as of attribute three. All the remaining three clusters (C_2, C_3, C_4) were created with similar procedures but with different parameters. The properties of each cluster are presented in Table 2. We ran KMCMD algorithm with random initial partition 50 times and studied the final clustering results. Final clustering results were inconsistent in different runs. Mostly, data points were clustered in two, three or four clusters. Examples of these clustering results are given in Tables 3–5. It is to be noted that in two different clustering runs there is no direct correspondence between final cluster labels. For example, C_{f1} (cluster label) in one instance of clustering may not be C_{f1} (cluster label) in the other instance of clustering.

Clustering results (Table 6) demonstrate that in contrast to KMCMD with the random partition, KMCMD with the *initKmix* algorithm was able to identify four clusters for the dataset accurately (AC = 0.908, RI

Table 2
Description of the clusters of the simulated dataset.

Cluster	Attributes			
	Categorical-1	Categorical-2	Numeric-1	Numeric-2
C_1	A (90%), B (10%)	X (90%), Y (10%)	$\mathcal{N}(-5,1)$	$\mathcal{N}(-5,1)$
C_2	A (90%), B (10%)	X (90%), Y (10%)	$\mathcal{N}(5,1)$	$\mathcal{N}(5,1)$
C_3	A (10%), B (90%)	X (10%), Y (90%)	$\mathcal{N}(-5,1)$	$\mathcal{N}(-5,1)$
C_4	A (10%), B (90%)	X (90%), Y (10%)	$\mathcal{N}(5,1)$	$\mathcal{N}(5,1)$

Table 3
One of the clustering results by the KMCMD algorithm with random initial partition. Most of the data points are in two clusters.

Original cluster	Final clusters			
	C_{f1}	C_{f2}	C_{f3}	C_{f4}
C_1	10	0	90	0
C_2	0	90	0	10
C_3	7	0	93	0
C_4	0	90	0	10

Table 4
One of the clustering results by the KMCMD algorithm with random initial partition. All the data points are in three clusters.

Original cluster	Final clusters		
	C_{f1}	C_{f2}	C_{f3}
C_1	0	82	18
C_2	18	82	0
C_3	0	0	100
C_4	100	0	0

Table 5
One of the clustering results by the KMCMD algorithm with random initial partition. The clustering correctly produced four almost equal sized clusters.

Original cluster	Final clusters			
	C_{f1}	C_{f2}	C_{f3}	C_{f4}
C_1	0	82	0	18
C_2	11	0	89	0
C_3	0	0	0	100
C_4	91	0	9	0

Table 6
Clustering results by the KMCMD algorithm with the *initKmix* algorithm. The final clustering has four quite accurate clusters.

Original cluster	Final clusters			
	C_{f1}	C_{f2}	C_{f3}	C_{f4}
C_1	90	0	0	10
C_2	0	89	11	0
C_3	7	0	0	93
C_4	0	9	91	0

Table 7
Description of the categorical datasets used in the experiments.

Dataset	Number of data points	Number of attributes	Number of classes
Soybean-small	47	35	4
Vote	435	16	2
Breast cancer	699	9	2
Mushroom	8124	22	2

Table 8
The clustering results (AC) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for categorical datasets. The better clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{AC}$	SD_{AC}	$\overline{AC}$	SD_{AC}
Soybean-small	**1**	0	0.967	0.079
Vote	**0.873**	0	0.871	0.002
Breast cancer	**0.974**	0	0.965	0.014
Mushroom	**0.894**	0	0.822	0.124

Table 9
The clustering results (RI) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for categorical datasets. The better clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{RI}$	SD_{RI}	$\overline{RI}$	SD_{RI}
Soybean-small	**1**	0	0.953	0.083
Vote	**0.779**	0	0.775	0.003
Breast cancer	**0.950**	0	0.931	0.016
Mushroom	**0.811**	0	0.708	0.147

Table 10
The clustering results (ARI) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for categorical datasets. The better clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{ARI}$	SD_{ARI}	$\overline{ARI}$	SD_{ARI}
Soybean-small	**1**	0	0.873	0.091
Vote	**0.557**	0	0.550	0.004
Breast cancer	**0.899**	0	0.861	0.017
Mushroom	**0.622**	0	0.415	0.163

= 0.916 and ARI = 0.775). We would like to emphasize that *initKmix* algorithm generates only one clustering result. This further highlights that application of *initKmix* algorithm produces accurate and consistent clustering results.

4.2. Categorical datasets

We carried out an experiment with four categorical datasets; Soybean-small, Vote, Breast cancer and Mushroom (Dua & Graff, 2017). Information on these datasets is provided in Table 7. We apply the KMCMD algorithm on those datasets. The numeric part of the distance measure presented in Eq. (2) will be zero in this case. The clustering results of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random partition method using different measures (AC, RI and ARI) using are presented in Tables 8–10. Unpaired t-test with 95% confidence interval (Freedman et al., 2007) was carried out to compare the performance of the KMCMD algorithm with the *initKmix* algorithm and the KMCMD algorithm with the random partition method. The calculations indicated that for all categorical datasets the *initKmix* algorithm performed statistically better than the random partition method.

The performance (AC) of the KMCMD algorithm with *initKmix* algorithm was also compared with k-modes algorithm with different state-of-the-art initialization methods; including Wu's initialization (Wu et al., 2007), Cao's initialization (He, 2006), Khan and Ahmad's initialization (Khan & Ahmad, 2013) and Ini_Entropy (Liu et al., 2016). The KMCMD algorithm with the *initKmix* algorithm was also compared with the Fuzzy k-modes clustering algorithm (Huang & Ng, 1999) and CRAFTER (Lin et al., 2018) algorithm. The results of various clustering methods are presented in Table 11. The results for the other clustering algorithms were taken from existing papers (Khan & Ahmad, 2013; Lin et al., 2018; Liu et al., 2016; Zhu & Xu, 2018). Except for the Soybean-small dataset, the combination of the KMCMD algorithm and *initKmix* outperformed the other clustering methods. For the Soybean-small dataset, the KMCMD algorithm with the *initKmix* algorithm performed similarly or better than the other clustering methods.

Table 11
Results (AC) for various clustering algorithms for categorical datasets."–" denotes results for a given algorithm that could not be obtained from the literature. The best clustering results are shown in bold.

Dataset	The KMCMD algorithm with *initKmix*	k-modes (Huang, 1997b) with random initialization	k-modes with Wu's initialization (Wu et al., 2007)	k-modes with Cao's initialization (He, 2006)	k-modes with Khan and Ahmad's initialization (Khan & Ahmad, 2013)	k-modes with Ini_Entropy initialization (Liu et al., 2016)	CRAFTER (Lin et al., 2018)	Fuzzy k-modes clustering (Huang & Ng, 1999)
Soybean-small	**1**	0.864	**1.000**	**1.000**	0.957	**1.000**	**1.000**	0.824
Vote	**0.873**	0.842	–	–	0.850	0.869	0.856	0.862
Breast cancer	**0.974**	0.836	0.911	0.911	0.913	0.933	–	–
Mushroom	**0.894**	0.875	0.875	0.875	0.882	0.888	0.774	0.723

Table 12
Description of the mixed datasets used in the experiments.

Dataset	Number of data points	Number of categorical attributes	Number of numeric attributes	Number of classes
Acute Inflammations	120	5	1	2
Heart (Statlog)	270	7	6	2
Heart (Cleveland)	303	6	7	2
Australian credit	690	8	6	2
German credit	1000	13	7	2

Table 13
The clustering results (AC) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for mixed datasets. The better clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{AC}$	SD_{AC}	$\overline{AC}$	SD_{AC}
Acute inflammation	**0.823**	0	0.762	0.125
Heart (Statlog)	**0.817**	0	0.802	0.010
Heart (Cleveland)	**0.841**	0	0.834	0.005
Australian credit	**0.858**	0	0.829	0.118
German credit	**0.683**	0	0.678	0.004

Table 14
The clustering results (RI) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for mixed datasets. The better clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{RI}$	SD_{RI}	$\overline{RI}$	SD_{RI}
Acute inflammation	**0.709**	0	0.639	0.118
Heart (Statlog)	**0.679**	0	0.665	0.013
Heart (Cleveland)	**0.728**	0	0.719	0.005
Australian credit	**0.756**	0	0.716	0.127
German credit	**0.567**	0	0.563	0.003

Table 15
The clustering results (ARI) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for mixed datasets. The better clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{ARI}$	SD_{ARI}	$\overline{ARI}$	SD_{ARI}
Acute inflammation	**0.415**	0	0.273	0.182
Heart (Statlog)	**0.358**	0	0.331	0.021
Heart (Cleveland)	**0.456**	0	0.438	0.009
Australian credit	**0.512**	0	0.432	0.217
German credit	**0.0519**	0	0.0415	0.006

4.3. Mixed datasets

The following five mixed datasets (Dua & Graff, 2017) were used in the experiments: Acute Inflammations, Heart (Statlog), Heart (Cleveland), Australian credit and German credit. Table 12 displays information on these datasets. The clustering results by the KMCMD algorithm with the *initKmix* algorithm and the KMCMD algorithm with the random partition method using different measures (AC, RI and ARI) using are presented in Tables 13–15. Unpaired t-test with 95% confidence interval (sample sizes were 50) was carried out to compare the performance of KMCMD algorithm with *initKmix* and the KMCMD algorithm with random initial partition. Results suggest that except Australian credit dataset for AC performance measure (Table 13), the *initKmix* algorithm performed statistically better than the random partition method for all other datasets on all the performance measures. For Australian credit dataset, there is no statistically significant difference *initKmix* between the performances of two methods. However, *initKmix* algorithm produced consistent clustering.

The performance of the KMCMD algorithm with the *initKmix* algorithm was also compared using AC performance measure with k-prototypes (Huang, 1997a) with random initialization, k-prototypes (Huang, 1997a) with the Ji et al. (Ji et al., 2015b) initialization method, Similarity-based Agglomerative clustering (SBAC) (Li & Biswas, 2002), Object-cluster similarity metric (OCIL) algorithm (Cheung & Jia, 2013) and fuzzy k-prototypes clustering (Ji et al., 2012). The results of various clustering methods are presented in Table 16. The results for the other clustering algorithms were taken from published papers (Cheung & Jia, 2013; Du et al., 2017; Ji et al., 2012). Except for the German credit dataset, the KMCMD algorithm with *initKmix* performed better than the other clustering algorithms. For the German credit dataset, the OCIL algorithm performed better than the KMCMD algorithm with *initKmix*.

4.4. Discussion

Wilcoxon Signed-Ranks test (Wilcoxon, 1945) with 95% confidence level was carried out to compare the performance of the KMCMD algorithm with the *initKmix* algorithm against KMCMD algorithm with random initial partition over all the nine datasets. The test suggests that the KMCMD algorithm with the *initKmix* algorithm significantly better than KMCMD algorithm with the random initial partition method.

The KMCMD algorithm with *initKmix* perform similar to or better than the other state-of-the-art clustering algorithms for categorical datasets. Some of these clustering methods use different initialization methods (He, 2006; Khan & Ahmad, 2013; Wu et al., 2007), and the better clustering results with the *initKmix* algorithm suggests that the it produces good initial partition. The similar behaviour is observed for mixed datasets. The proposed approach has the best performance across clustering methods for mixed datasets for four out of five datasets. One of these clustering methods (Ji et al., 2015b) uses an initialization method, but the better clustering results point to the superiority of the *initKmix* algorithm in creating an initial partition.

In *initKmix* algorithm, we get initial partition of a dataset for the KMCMD algorithm after combining many clustering results. It is possible that this initial partition can be used as the final clustering result. However, the goal of this paper is to study the performance of a KMCMD algorithm with initial partition created by *initKmix* algorithm.

Table 16
Results (AC) for various clustering algorithms for mixed datasets. "–" denotes results for a given algorithm that could not be obtained from the literature. The best clustering result is shown in bold.

Dataset	The KMCMD algorithm with *initKmix*	k-prototypes (Huang, 1997a) with random initialization	k-prototypes (Huang, 1997a) with Ji et al. (2015b) initialization method	Similarity-based Agglomerative clustering (SBAC) (Li & Biswas, 2002)	Object-cluster similarity metric (OCIL) algorithm (Cheung & Jia, 2013)	Fuzzy k-prototypes (Ji et al., 2012)
Acute inflammation	**0.823**	0.610	–	0.508	–	0.710
Heart (Statlog)	**0.817**	0.770	–	–	0.814	–
Heart (Cleveland)	**0.841**	0.772	0.808	0.752	0.831	0.835
Australian credit	**0.858**	0.738	0.800	0.600	0.757	0.838
German credit	0.683	0.671	–	–	**0.695**	–

The results suggest that the KMCMD algorithm with the *initKmix* algorithm produces accurate clustering for both categorical and mixed datasets. The *initKmix* algorithm generates the accurate initial partition, which in turn improves the performance of KMCMD in comparison to random initial partition. The accurate and diverse clustering results are the key for an accurate cluster ensemble (Strehl & Ghosh, 2003). Accurate initial partition suggests that the *initKmix* algorithm is able to create accurate and diverse clustering results in different runs. Using each attribute for creating initial clusters in different runs of the KMCMD algorithm could be the reason for it.

4.5. Effect of k on the performance of the KMCMD algorithm with the initkmix algorithm

In the experiments (Sections 4.1–4.3) for each dataset, the number of the desired clusters (k) was equal to the number of the actual classes. We also carried out experiments to observe the performance of the KMCMD algorithm with the *initKmix* algorithm when the number of desired clusters was not equal to the number of actual classes. We selected a mixed dataset, Australian credit, for our experiment. The number of the actual classes was two for the dataset. We carried out experiments with k = 4, 5, 6 and 7. The results are presented in Tables 17–19. For AC performance measures, we did not observe huge variation as the difference between maximum AC (0.858 for k = 2) and minimum AC (0.823, for k = 6) is 0.035, which is around 4% of the maximum value (as shown in Table 17). The results demonstrate that the performance of KMCMD with the *initKmix* algorithm is robust to the value of k.

However, the differences between maximum and minimum values are quite large for RI measure (21.56% of the maximum value) and ARI measure (57.18% of the maximum value). That was expected as the number of the clusters increases, the data points in a original class tend to be in different clusters, which leads to lower values of these performance measures.

We also compared the *initKmix* algorithm against the random initial partition for different values of k. Experimental set up was the same as discussed at the start of Section 4. The average and standard deviation of various performance measures are presented in Tables 17–19. Expect in one case (k = 6 and AC performance measure), KMCMD with the *initKmix* algorithm outperformed KMCMD with the random initial partition in all the cases.

The results show that *initKmix* initialization is effective even if k is not equal to the actual number of clusters.

Table 17
The clustering results (AC) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for Australian credit datasets for different numbers of k. The better clustering result is shown in bold.

k	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{AC}$	SD_{AC}	$\overline{AC}$	SD_{AC}
2	**0.858**	0	0.829	0.118
4	**0.849**	0	0.837	0.010
5	**0.852**	0	0.833	0.009
6	0.823	0	**0.835**	0.011
7	**0.839**	0	0.832	0.025

Table 18
The clustering results (RI) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for Australian credit datasets for different numbers of k. The better clustering result is shown in bold.

k	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
	$\overline{RI}$	SD_{RI}	$\overline{RI}$	SD_{RI}
2	**0.756**	0	0.716	0.127
4	**0.682**	0	0.651	0.007
5	**0.685**	0	0.648	0.019
6	**0.616**	0	0.612	0.022
7	**0.593**	0	0.584	0.025

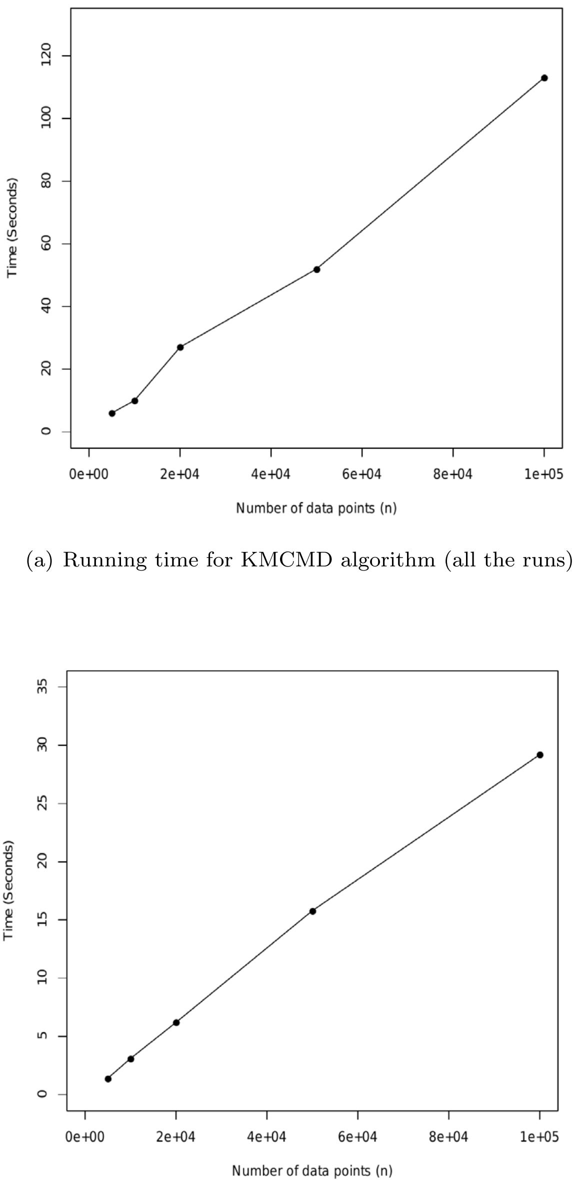
Table 19
The clustering results (ARI) of the KMCMD algorithm with *initKmix* and the KMCMD algorithm with the random initial partition method for Australian credit datasets for different numbers of k. The better clustering result is shown in bold.

k	The KMCMD algorithm with *initKmix*		The KMCMD algorithm with the random initial partition	
2	**0.512**	0	0.432	0.217
4	**0.368**	0	0.307	0.014
5	**0.372**	0	0.306	0.038
6	**0.237**	0	0.228	0.044
7	**0.216**	0	0.191	0.049

4.6. Effect of n and k on the running time

We carried out experiments with different datasets to study the effect of n and k on the running time of the KMCMD algorithm with the *initKmix* algorithm. The experiments were done on a computer with Intel Core i7 1.80 GHz and 16 GB RAM. To study the effect of n on the running time, five artificial datasets of different sizes (5000, 10 000, 20 000, 50 000 and 100 000) were generated (by *clustMixType* package (Szepannek & Aschenbruck, 2020)) using the same procedure as discussed in Section 4.1. These datasets had four attributes and the value of k was set to four. As there were two different parts of the *initKmix* algorithm (different runs of KMCMD algorithm and cluster ensemble algorithms) and their implementation is in different programming platforms, we present their results in different figures for better understanding. The results presented in Figs. 2(a) and 2(b) demonstrate that the running times of both parts of the *initKmix* algorithm are linear with respect to n.

We also carried experiments to study the effect of the value of k on the running time of *initKmix* algorithm. We selected the dataset with 100,000 data points for the experiment. Four different values of k (2, 4, 6 and 8) were used in the experiment. The running times for KMCMD algorithm (for all runs) and cluster ensemble algorithm are presented in Figs. 3(a) and 3(b). The results suggest that there is a increase in the running time of KMCMD algorithm; however, no significant changes were observed for cluster ensemble algorithms. We can infer from these running times that the total time of *initKmix* algorithm increases with the value of k. It is to be noted that we run KMCMD algorithm m time



(a) Running time for KMCMD algorithm (all the runs)

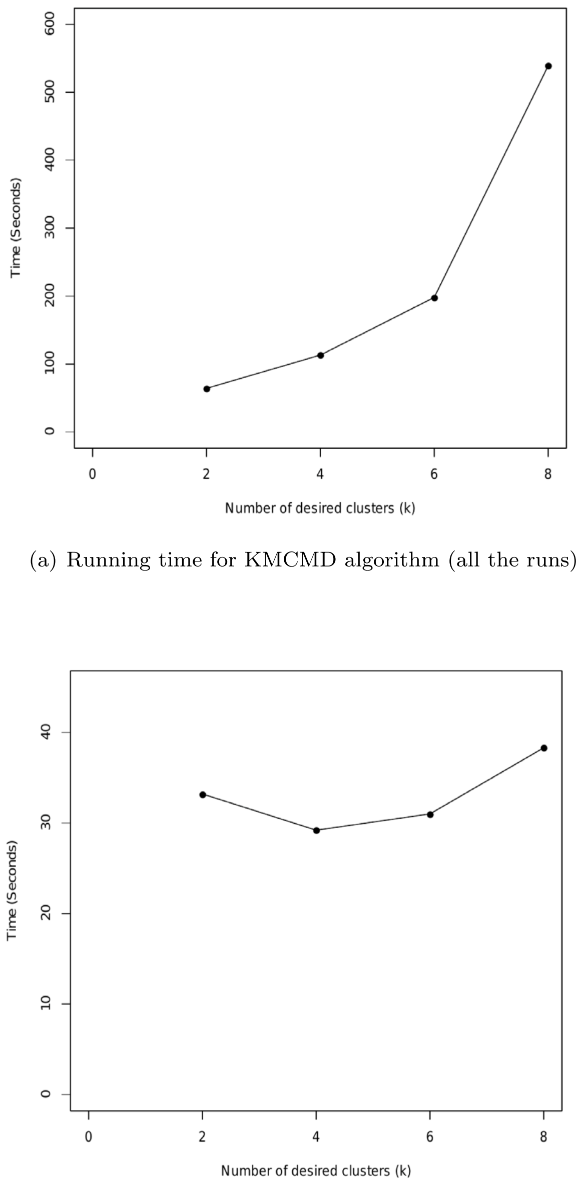
(b) Running time for combining clustering results

Fig. 2. Running time vs the number of data points graphs. The (a) figure presents the time taken by KMCMD algorithm. The (b) figure shows the time taken to combine the clustering results.

using each attribute as initial clusters, the value of desired clusters in a run with a categorical attribute is set to the number of unique attribute values. Therefore, it is independent of the value of k. In the present case, two out of four instances of clustering results are the same for all the values of k.

4.7. Analysis of individual clustering results

Regarding the KMCMD algorithm with the *initKmix* algorithm, we run the KMCMD algorithm m times to produce m clustering results. These m results are combined to yield the initial partition and then the KMCMD algorithm is run with this initial partition to obtain the final clustering results. Therefore, we perform an analysis to compare the accuracy of m individual clustering results and the final clustering result using the AC performance measure.



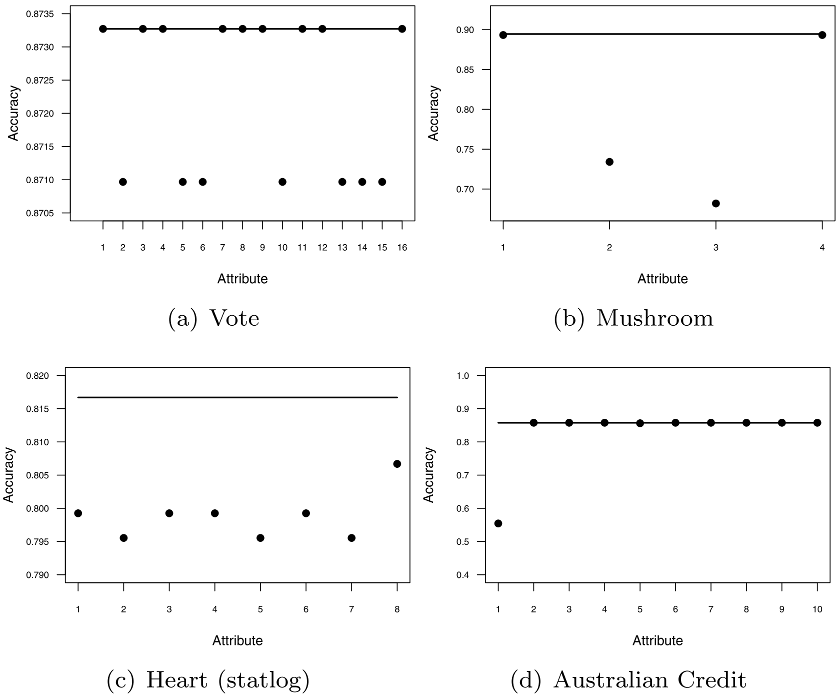
(a) Running time for KMCMD algorithm (all the runs)

(b) Running time for combining clustering results

Fig. 3. Running time vs the number of clusters (k) graphs. The (a) figure presents the time taken by KMCMD algorithm. The (b) figure shows the time taken to combine the clustering results.

For this analysis, we selected two categorical datasets, Vote and Mushroom, and two mixed datasets, Heart (Statlog) and Australian credit, for this analysis. For better comparative study, for categorical datasets, we selected these datasets with some attributes having the same number of values as the number of desired clusters. Similarly, for mixed datasets, we selected these datasets with some categorical attributes having the same number of values as the number of desired clusters. Using categorical attributes, with the numbers of attribute values not equal to the desired clusters, to create the initial clusters does not produce the desired number of clusters. Therefore, clustering results when those attributes were used as initial clusters were not selected for the comparative study.

For Vote dataset, the individual clustering results for 16 categorical attributes as initial clusters in different runs are presented in Fig. 4(b). We did not observe large differences in individual clustering results (the minimum AC - 0.8709 and the maximum AC - 0.8732). The final AC



(a) Vote

(b) Mushroom

(c) Heart (statlog)

(d) Australian Credit

Fig. 4. Clustering results for various datasets. A dark filled circular dot shows the performance for the KMCMD algorithms by using that attribute for obtaining initial partition. The straight line represents the final clustering result obtained with the *initKmix* algorithm. The dark filled circular dot below the line shows that the performance in that run is worse than the final result. The dark filled circular dot on the line shows that the performance in that run is similar to the final result.

(0.8732) was equal to the maximum AC. For the Mushroom dataset, only four attributes had two values (same as the number of desired clusters). The results for these four attributes are presented in Fig. 4(b). There was a large variation in individual clustering results (the minimum AC - 0.682 , the maximum AC - 0.893). The final clustering result (0.894) was slightly better than the best individual clustering (0.893). Fig. 4(c) has individual clustering results for the *Heart (Statlog)* dataset (the minimum AC - 0.795, the maximum AC - 0.807), the final clustering accuracy was 0.817, which was slightly better than the best individual accuracy (0.807). The individual clustering results for *Australia credit* dataset (for six numeric and four categorical) are presented in Fig. 4(d) (the minimum AC - 0.554 , the maximum AC - 0.858). There was a large variation in individual clustering results. The final clustering result (0.858) was equal to the best individual clustering result.

The analysis suggests that individual clustering results for various datasets had small or large variations; however, the final clustering results were equal to or better than the best individual clustering results. This shows that combining clustering results is a good approach to obtain better initial clusters, and when feeding these to the KMCMD algorithm results in better clustering accuracy for the studied categorical and mixed datasets.

5. Conclusion and future work

KMD algorithms suffer from the random initial partition problem that can lead to different clustering results in different runs thereby undermining the reliability of results. In this paper, we presented *initKmix,* an algorithm to find the initial partition for the KMCMD algorithm (Ahmad & Dey, 2007). The algorithm uses an individual attribute to create the initial partition when running the KMCMD algorithm. Multiple clustering results created by this procedure are combined to obtain the initial partition. The clustering results obtained using the KMCMD algorithm with the initial partition created by *initKmix* were accurate and consistent. The KMCMD algorithm with *initKmix* algorithm outperformed KMCMD algorithm with the random initial partition method on multiple categorical and mixed datasets. The KMCMD algorithm with *initKmix* algorithm also performed similar to or better than other state-of-the-art clustering algorithms on multiple categorical and mixed datasets. Computational complexity analysis and running time results suggest that the running time of the KMCMD algorithm with *initKmix* is linear with respect to number of data points. Therefore, this clustering algorithm scales well with the number of data points. Results also demonstrated that the performance of the KMCMD algorithm with the *initKmix* algorithm is robust to the choice of the k.

In future, other KMD algorithms (Huang, 1997a; Huang et al., 2005; Modha & Spangler, 2003) with *initKmix* will also be studied. Fuzzy clustering (D'Urso, 2015) and subspace clustering (Parsons et al., 2004) are important research areas in clustering as they capture different cluster structures than those captured by hard clustering. KMD algorithms have been suggested for fuzzy clustering (Du et al., 2017; Ji et al., 2012) and subspace clustering (Ahmad & Dey, 2011), in future, we will consider applying the *initKmix* algorithm to these clustering algorithms. We will also investigate possible extension of *initKmix* to find the value of k for KMD algorithms.

CRediT authorship contribution statement

Amir Ahmad: Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Data curation, Supervision, Writing - original draft, Visualization, Funding acquisition. **Shehroz S. Khan:** Conceptualization, Methodology, Validation, Formal analysis, Writing - review & editing, Visualization.

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.

Funding

This research was funded by a UAE university Start-up grant (grant number G00002668; fund number 31T101).

References

Ahmad, A., & Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data and Knowledge Engineering*, *63*(2), 503–527.

Ahmad, A., & Dey, L. (2011). A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. *Pattern Recognition Letters*, *32*(7), 1062–1069.

Ahmad, A., & Hashmi, S. (2016). K-harmonic means type clustering algorithm for mixed datasets. *Applied Soft Computing*, *48*(C), 39–49.

Ahmad, A., & Khan, S. S. (2019). Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access*, *7*, 31883–31902.

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms* (pp. 1027–1035). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Balaji, K., & Lavanya, K. (2018). Clustering algorithms for mixed datasets: A review. *International Journal of Pure and Applied Mathematics*, *18*(7), 547–556.

Bishop, C. M. (2008). *Pattern recognition and machine learning*. Springer-Verlag New York Inc.

Bradley, P. S., & Fayyad, U. M. (1998). Refining initial points for k-means clustering. In *ICML 1998 proceedings of the fifteenth international conference on machine learning* (pp. 91–99). Morgan kaufmann.

Chen, J., Xiang, X. L., Zheng, H., & Bao, X. (2017). A novel cluster center fast determination clustering algorithm. *Applied Soft Computing*, *57*, 539–555.

Cheung, Y. M., & Jia, H. (2013). Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. *Pattern Recognition*, *46*(8), 2228–2238.

Du, M., Ding, S., & Xue, Y. (2017). A novel density peaks clustering algorithm for mixed data. *Pattern Recognition Letters*, *97*, 46–53.

Dua, D., & Graff, C. (2017). *UCI machine learning repository*. University of California, Irvine, School of Information and Computer Sciences, URL http://archive.ics.uci.edu/ml.

Duda, R. O., Hart, P. E., & Stork, D. G. (1973). *Pattern classification and scene analysis*. Wiley.

D'Urso, P. (2015). Fuzzy clustering. In C. Hennig, M. Meila, F. Murtagh, & R. Rocci (Eds.), *Handbook of cluster analysis* (1st ed.). (pp. 545–573). Chapman and Hall/CRC, The address of the publisher. chapter 24.

D'Urso, P., & Massari, R. (2019). Fuzzy clustering of mixed data. *Information Sciences*, *505*, 513–534.

Freedman, D., Pisani, R., & Purves, R. (2007). *Statistics*. W. W. Norton & Company.

Ghosh, J., & Acharya, A. (2011). Cluster ensembles. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, *1*(4), 305–315.

He, Z. (2006). Farthest-point heuristic based initialization methods for K-modes clustering. CoRR abs/cs/0610043.

He, Z., Xu, X., & Deng, S. (2005). Clustering mixed numeric and categorical data: A cluster ensemble approach. eprint arXiv:cs/0509011.

Huang, Z. (1997a). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the first pacific asia knowledge discovery and data mining conference* (pp. 21–34). Singapore: World Scientific.

Huang, Z. (1997b). A fast clustering algorithm to cluster very large categorical data sets in data mining. In *In research issues on data mining and knowledge discovery* (pp. 1–8).

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, *2*(3), 283–304.

Huang, Z., & Ng, M. K. (1999). A fuzzy k-modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems*, *7*(4), 446–452.

Huang, J. Z., Ng, M. K., Rong, H., & Li, Z. (2005). Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(5), 657–668.

Iam-On, N., Boongeon, T., Garrett, S., & C. Price, C. (2012). A link-based cluster ensemble approach for categorical data clustering. *IEEE Transactions on Knowledge and Data Engineering*, *24*(3), 413–425.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data*. Prentice Hall.

Ji, J., Pang, W., Zheng, Y., Wang, Z., & Ma, Z. (2015). An initialization method for clustering mixed numeric and categorical data based on the density and distance. *International Journal of Pattern Recognition and Artificial Intelligence*, *29*(7), Article 1550024.

Ji, J., Pang, W., Zheng, Y., Wang, Z., Ma, Z., & Zhang, L. (2015a). A novel cluster center initialization method for the k-prototypes algorithms using centrality and distance. *Applied Mathematics & Information Sciences*, *9*(6), 2933–2942.

Ji, J., Pang, W., Zheng, Y., Wang, Z., Ma, Z., & Zhang, L. (2015b). A novel cluster center initialization method for the k-prototypes algorithms using centrality and distance. *Applied Mathematics and Information Sciences*, *9*(6), 2933.

Ji, J., Pang, W., Zhou, C., Han, X., & Wang, Z. (2012). A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. *Knowledge-Based Systems*, *30*, 129–135.

Kashan, A. H. (2009). League championship algorithm: A new algorithm for numerical function optimization. In *2009 International conference of soft computing and pattern recognition*. (pp. 43–48).

Khan, S. S., & Ahmad, A. (2003). Computing initial points using density based multiscale data condensation for clustering categorical data. In *2nd International conference on applied artificial intelligence, (vol. 3)*.

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

Khan, S. S., & Ahmad, A. (2013). Cluster center initialization algorithm for K-modes clustering. *Expert Systems with Applications*, *40*(18), 7444–7456.

Khan, S. S., & Kant, S. (2007). Computation of initial modes for k-modes clustering algorithm using evidence accumulation. In *Proceedings of the 20th international joint conference on artifical intelligence* (pp. 2784–2789). Morgan Kaufmann Publishers Inc.

Li, C., & Biswas, G. (2002). Unsupervised learning with mixed numeric and nominal data. *IEEE Transaction on Knowledge and Data Engineering*, *14*(4), 673–690.

Liang, J., Zhao, X., Li, D., Cao, F., & Dang, C. (2012). Determining the number of clusters using information entropy for mixed data. *Pattern Recognition*, *45*(6), 2251–2265.

Lin, S., Azarnoush, B., & Runger, G. (2018). CRAFTER: A tree-ensemble clustering algorithm for static datasets with mixed attributes and high dimensionality. *IEEE Transactions on Knowledge and Data Engineering*, *30*(9), 1686–1696. http://dx.doi.org/10.1109/TKDE.2018.2807444.

Liu, F. J. G., Du, J., & Sui, Y. (2016). Initialization of K-modes clustering using outlier detection techniques. *Information Sciences*, *332*(C), 167–183.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam, & J. Neyman (Eds.), *Proc. of the fifth berkeley symposium on mathematical statistics and probability, (vol. 1)* (pp. 281–297). University of California Press.

Modha, D. S., & Spangler, W. S. (2003). Feature weighting in k-means clustering. *Machine Learning*, *52*(3), 217–237.

Muller, E., Gunnemann, S., Farber, I., & Seidl, T. (2012). Discovering multiple clustering solutions: Grouping objects in different views of the data. In *2012 IEEE 28th international conference on data engineering*. (pp. 1207–1210).

Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter*, *6*(1), 90–105.

R Core Team (2013). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, URL http://www.R-project.org/.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, *66*(336), 846–850.

Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492–1496.

Strehl, A. (2011). Cluster analysis and cluster ensemble software. URL http://strehl.com/soft.html. (Accessed 2 July 2020).

Strehl, A., & Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal Machine Learning Research*, *3*, 583–617.

Szepannek, G., & Aschenbruck, R. (2020). Clustmixtype - k-prototypes clustering for mixed variable-type data. https://CRAN.R-project.org/package=dplyr. R package version 0.2-5.

Topchy, A., Jain, A. K., & Punch, W. (2005). Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(12), 1866–1881.

Wangchamhan, T., Chiewchanwattana, S., & Sunat, K. (2017). Efficient algorithms based on the k-means and Chaotic League Championship Algorithm for numeric, categorical, and mixed-type data clustering. *Expert Systems with Applications*, *90*, 146–167.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). CA: Morgan Kaufmann San Francisco.

Wu, S., Jiang, Q., & Huang, J. Z. (2007). A new initialization method for clustering categorical data. In Z.-H. Zhou, H. Li, & Q. Yang (Eds.), *Advances in knowledge discovery and data mining* (pp. 972–980). Berlin, Heidelberg: Springer Berlin Heidelberg.

Yang, Y., & Wang, H. (2018). Multi-view clustering: A survey. *Big Data Mining and Analytics*, *1*(2), 83–107.

Zhang, B. (2001). Generalized k-harmonic means–dynamic weighting of data in unsupervised learning. In *Proceedings of the 2001 SIAM international conference on data mining* (pp. 1–13). SIAM.

Zheng, Z., Gong, M., Ma, J., Jiao, L., & Wu, Q. (2010). Unsupervised evolutionary clustering algorithm for mixed type data. In *IEEE congress on evolutionary computation*. (pp. 1–8).

Zhu, S., & Xu, L. (2018). Many-objective fuzzy centroids clustering algorithm for categorical data. *Expert Systems with Applications*, *96*, 230–248.