The k-means algorithm: A comprehensive survey and performance evaluation

Mohiuddin Ahmed  
*Edith Cowan University*

Raihan Seraj

Syed Mohammed Shamsul Islam  
*Edith Cowan University*

Follow this and additional works at: [https://ro.ecu.edu.au/ecuworkspost2013](https://ro.ecu.edu.au/ecuworkspost2013)  

[10.3390/electronics9081295](https://doi.org/10.3390/electronics9081295)


This Journal Article is posted at Research Online.  
The k-means Algorithm: A Comprehensive Survey and Performance Evaluation

Mohiuddin Ahmed 1,* , Raihan Seraj 2 and Syed Mohammed Shamsul Islam 1,3

1 School of Science, Edith Cowan University, Joondalup 6027, Australia; syed.islam@ecu.edu.au
2 Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 0G4, Canada; raihan.seraj@mail.mcgill.ca
3 School of Computer Science and Software Engineering, The University of Western Australia, Crawley 6009, Australia
* Correspondence: mohiuddin.ahmed@ecu.edu.au

Received: 29 May 2020; Accepted: 7 August 2020; Published: 12 August 2020

Abstract: The k-means clustering algorithm is considered one of the most powerful and popular data mining algorithms in the research community. However, despite its popularity, the algorithm has certain limitations, including problems associated with random initialization of the centroids which leads to unexpected convergence. Additionally, such a clustering algorithm requires the number of clusters to be defined beforehand, which is responsible for different cluster shapes and outlier effects. A fundamental problem of the k-means algorithm is its inability to handle various data types. This paper provides a structured and synoptic overview of research conducted on the k-means algorithm to overcome such shortcomings. Variants of the k-means algorithms including their recent developments are discussed, where their effectiveness is investigated based on the experimental analysis of a variety of datasets. The detailed experimental analysis along with a thorough comparison among different k-means clustering algorithms differentiates our work compared to other existing survey papers. Furthermore, it outlines a clear and thorough understanding of the k-means algorithm along with its different research directions.

Keywords: clustering; k-means; initialization; categorical attributes; cyber security; healthcare; unsupervised learning

1. Introduction

The advancements in computing, along with the rapid growth and availability of data repositories, have often emphasized the task of gaining meaningful insights from the data. This encourages taking appropriate measures based on knowledge discovery approaches. Machine Learning (ML) can be broadly classified into two: supervised and unsupervised learning. In the case of supervised learning, a function is learned that maps a given input to an output based on the available input–output pairs. These learning algorithms thus require the availability of labeled data with the desired output value [1]. The availability of labeled data represents an ideal scenario; however, such datasets are often expensive and challenging to obtain. For instance, in the intrusion detection domain, zero-day attacks are rare instances and obtaining labels for them is expensive. Hence, when the labels of the datasets are unavailable, unsupervised learning approaches are typically used [2]. Under such a learning framework, the algorithm has no prior knowledge of the true labels of the dataset and tries to draw inferences from the dataset itself. A more recent and popular set of algorithms referred to as semi-supervised learning consists of algorithms that lie in between supervised and unsupervised learning. Such a learning framework makes use of labeled and unlabeled data for better inference. Existing research on semi-supervised learning corroborates that adding a skimpy amount of labeled
data together with a large amount of unlabeled data produces considerable improvements in the accuracy of the predictive model. This manuscript primarily deals with different variations of the \textit{k-means} algorithm, which falls under the family of unsupervised learning. Therefore, this paper will focus only on unsupervised learning algorithms.

Clustering algorithms exploit the underlying structure of the data distribution and define rules for grouping the data with similar characteristics \cite{3}. This process results in the partition of a given dataset according to the clustering criteria without any prior knowledge about the dataset. In an ideal clustering scenario, each cluster consists of similar data instances that are quite dissimilar from the instances in other clusters. Such a dissimilarity measure relies on the underlying data and objective of the algorithm. Clustering is central to many data-driven applications and is considered an interesting and important task in machine learning. It is also studied in statistics, pattern recognition, computational geometry, bioinformatics, optimization, image processing, and in a variety of other fields \cite{4–7}. A plethora of clustering techniques have been invented in the last decade, which are being applied in a wide range of application domains \cite{8}. Table 1 shows the recent applications on \textit{k-means} clustering \cite{9} in different application domains.

\begin{table}[h]
\centering
\caption{Applications of variants of the \textit{k-means} algorithm in different application domains.}
\begin{tabular}{lll}
\hline
Reference & Application & Algorithm \\
\hline
\cite{10} & Face detection & Symmetry-based version of \textit{k-means} (S8KM). \\
\cite{11} & Mobile storage positioning & Potential \textit{k-means}. \\
\cite{12} & Load pattern & Hierarchical \textit{k-means} (H-Kmeans). \\
\cite{13} & Wireless sensor networks & Distributed \textit{k-means} and fuzzy \textit{c-means}. \\
\cite{14} & Partial multiview data & Weighted \textit{k-means}. \\
\cite{15} & Mobile health & \textit{k-means} implemented with CORDIC. \\
\cite{16} & Endpoint detection & \textit{k-means} for realtime detection. \\
\cite{17} & Big data & Privacy preserving \textit{k-means}. \\
\cite{18} & Multiview data & \textit{k-means}. \\
\cite{19} & Wind power forecasting & \textit{k-means} with bagging neural network. \\
\cite{20} & Social tags & \textit{k-means} based on latent semantic analysis. \\
\cite{21} & Sensing for IGBT current & \textit{k-means} with neural network. \\
\cite{22} & Image segmentation. & Kernel \textit{k-means} Nystrom approximation. \\
\cite{23} & Image compression & \textit{k-means} cuckoo optimization. \\
\cite{24} & Sound source angle estimation. & Neural network based on global \textit{k-means}. \\
\cite{25} & Shape recognition & Fuzzy \textit{k-means} clustering ensemble (FKMCE). \\
\cite{26} & Signal processing & Compressive \textit{k-means} clustering (CKM). \\
\cite{27} & Text processing & Vanilla \textit{k-means}. \\
\cite{28} & High dimensional data processing & Fast adaptive \textit{k-means} (FAKM). \\
\cite{29} & Computational complexity & Multiple kernel \textit{k-means} with late fusion. \\
\cite{30} & Image processing & A hybrid parallelization of \textit{k-means} algorithm. \\
\cite{31} & Adaptive clustering & Fuzzy \textit{k-means} with S-distance. \\
\cite{32} & DDoS detection & Semi-supervised \textit{k-means} algorithm with hybrid feature. \\
\cite{33} & Optimization & Non alternating stochastic \textit{k-means}. \\
\cite{34} & Data Summarization & Modified \textit{x-means}. \\
\hline
\end{tabular}
\end{table}
This survey studies the problems of and solutions to partition-based clustering, and more specifically the widely used \textit{k-means} algorithm [9], which has been listed among the top 10 clustering algorithms for data analysis [35]. Due to their popularity and ease of use, \textit{k-means} clustering methods are being used in conjunction with deep learning for tasks such as image segmentation and handwriting recognition [36]. A more recent work in [37] used a fully connected deep convolution neural network along with \textit{k-means} and performed pixel matching between a segmented image and a convoluted image. This overcomes the problem that exists when useful information from images is lost by repeated convolution of the images [36]. Although the \textit{k-means} clustering algorithm itself performs well with compact and hyper-spherical clusters, we are interested in highlighting its limitations and suggesting solutions. The primary focus is given to two unavoidable problems of the \textit{k-means} algorithm: (i) assignment of centroids and number of clusters and (ii) ability to handle different types of data.

Although researchers have proposed variants (Figure 1) of \textit{k-means} algorithms to overcome these impediments, they are however domain specific and do not generalize well. For example, a \textit{k-means} algorithm which can handle categorical data might perform poorly because of the initialization process used. Table 2 outlines a comparative analysis of existing surveys on clustering and highlights the contributions of this paper. The key contributions of this paper are listed below.

- Existing solutions of the \textit{k-means} algorithm along with a taxonomy are outlined and discussed in order to augment the understanding of these variants and their relationships.
- This research frames the problems, analyses their solutions and presents a concrete study on advances in the development of the \textit{k-means} algorithm.
- Experiments are performed using the improved \textit{k-means} algorithms to find out their effectiveness using six benchmark datasets.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{variants.png}
\caption{A simple taxonomy of variants of \textit{k-means} algorithm.}
\end{figure}

\textbf{Table 2.} Comparison with existing surveys.

<table>
<thead>
<tr>
<th>Survey</th>
<th>Initialization</th>
<th>Data Types</th>
<th>Applications</th>
<th>Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang [38]</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Filippone [39]</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Rai [40]</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>This paper</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

\textbf{Paper Roadmap}

Our paper is structured as follows. Sections 2 and 3 address the problems of the \textit{k-means} algorithm and review the existing solutions for them. Section 4 showcases an experimental analysis of the representative algorithms (from Sections 2 and 3) on six benchmark datasets widely used by the machine learning and data mining research community. Section 5 concludes the paper, summarizing the key insights regarding the \textit{k-means} algorithm.
2. *k*-means Variants for Solving the Problem of Initialization

The *k*-means algorithm depends on the value of *k*; which always needs to be specified in order to perform any clustering analysis. Clustering with different *k* values will eventually produce different results. Different initialization problems that were analyzed in recent studies did not consider the problem where the algorithm only converges to a poor local minima. In [41], an alternative approach was adopted to prevent the *k*-means algorithm from being easily affected by noise and outlier values. The authors presented a modified *k*-means algorithm based on self-paced learning theory. Self-paced learning theory is used in order to select competing training subset that helps the *k*-means algorithm to build an initial cluster model. The generalization ability of the algorithm is then improved by subsequently adding training subsets found by self-paced learning until the model reaches an optimal performance or all the training samples have been used. The authors proposed this algorithm and demonstrated its performances on several real datasets.

The authors in [42] proposed an improvement to the vanilla *k*-means algorithm that prevents it from getting stuck to a local minima. The improved algorithm incorporates cuckoo search along with the *k*-means algorithm. The method incorporates a modified cuckoo search algorithm, which helps to reach a better solution since the search step size factor is being changed. Overall, the proposed algorithm is robust as well as fast in reaching a near-optimal solution.

In [43], the authors discussed criteria concerning the stability and the performance of the *k*-means. Theoretical results of algorithm stability were proven as the number of instances approaches infinity. The authors in [44] analyzed the stability of the *k*-means clustering algorithm in a more practical scenario, the parameter (cluster number) is chosen by stability-based model selection. The primary focus was towards drawing random sample points to explain the effect on the stability of the algorithm.

An initial estimation method for computing the covariance matrix for the *k*-means algorithm using Mahalanobis distance was proposed in [45]. It involves finding a group of points comprised of neighbors with a high density that represent the centroids of the selected clusters. These provide an approximate estimate of the covariance matrix, which is updated successively using the proposed algorithm.

Ball et al. [46] proposed the ISODATA algorithm to estimate *k* by parameter tweaking. A similar strategy is also adopted in adaptive resonance theory [47], which generates new clusters [48]. Since this approach requires a predefined threshold, it is generally avoided.

In *x*-means clustering [49], the BIC (Bayesian information criterion) or “Schwarz criterion” [50,51] is utilised to find the number of *k* clusters. For a given set of data, a family of different models initialized with different *k* values is used to compute the posterior probability distribution. These distributions are then used to find the score of the different models. This algorithm was utilized in identifying anomalous patterns in [52,53].

Bradley et al. [54] outlined methods for identifying a more fine grained starting condition that relies on the estimation of the distribution modes. The algorithm initially chooses small samples from the dataset and applies *k*-means clustering. These temporary sets are then further clustered using the *k*-means to find better initialization. This algorithm’s computational complexity is significantly lower compared to vanilla *k*-means and, more importantly, is scalable to large datasets. Figure 2 displays the methodology for obtaining refined initial points.

In [55], the author empirically compared four initialization methods. These are as follows: (i) RANDOM, (ii) FA [56], (iii) MA [9], and (iv) KA [57]. The RANDOM method divides the dataset randomly into *k* number of clusters and is one of the most popular initialization methods. The FA approach [56] randomly chooses *k* instances from the given dataset and assigns the remaining instances to the nearest cluster centers. The MA approach [9] randomly selects *k* instances of the dataset, similar to basic *k*-means. The KA approach [57] provides a method where the initial clustering is done by selecting representative instances successively until *k* instances are found. The first representative instance is the most centrally located in the dataset. The remaining representative instances are chosen based on the heuristic that presents rules of choosing instances with higher numbers compared to
others. Based on their experimental analysis, the RANDOM and KA resulted in superior performance compared to the other two. Recently, in [58], a co-clustering algorithm is proposed for dealing with sparse and high dimensional data that outperforms the basic $k$-means.

![Figure 2. Refined initial instances, adapted from [54].](image)

These solutions can only handle numerical data. Hence such methods would not be suitable for datasets consisting of mixed attributes, for instance, categorical and binary. In the next section, we discuss the research progress dealing with mixed types of data for clustering.

### 3. $k$-means Variants for Solving the Problem of Data Issue

In many application domains, datasets include attributes that are of mixed data types [59]. Therefore, a distance calculation scheme is necessary that is able to compute distances between instances having mixed types of data. This section discusses existing research on $k$-means to find similarity or distance metrics between categorical attributes. Although the algorithms discussed in the previous section solve the initialization problem, they still rely on numerical data for distance calculation. Therefore, it is necessary to explore the distance calculation mechanism for the $k$-means clustering algorithm and our investigation as follows.

Wang et al. [60] introduced an extended form of the fuzzy $k$-means (xFKM) algorithm for datasets with non-numerical attributes, such as categorical data. The centroids constitute an extended form to retain as much clustering information as possible. A computational cost comparison was made among the $k$-means, fuzzy $k$-means and xFKM. They showed that the xFKM is most suitable for datasets with categorical attributes, where the attributes vary in a medium and acceptable range of diverse values.

Amir et al. [61] offered a cost function and distance measure for clustering datasets with mixed data (datasets with numerical and categorical data) based on co-occurrences of values. In [62], a kernel function based on “hamming distance” [62] was proposed for embedding categorical data. The kernel-$k$-means provides an add-on to the $k$-means clustering that is designed to find clusters in a feature space where distances are calculated via kernel functions.

Huang et al. in [63] provided a variant of the $k$-means algorithm, where a dissimilarity measure is used as a metric. The algorithm combines the $k$-means and $k$-modes clustering for data having mixed attributes. In [64], the convergence of different versions of the modified $k$-modes algorithms was analyzed. The authors proved that the modified algorithms fail to converge to a local minima without degrading the original $k$-mode algorithm. To handle this issue, the authors presented two algorithms, MKM-NOF and MKM-NDM, that apply different methods for representing clusters by weighted cluster prototypes.
In [65], an “ellipsoidal k-means” algorithm was proposed that extends the “spherical k-means” algorithm for selecting features in a sparse and high-dimensional dataset. In [66], an entropy weighting k-means (EWKM) clustering algorithm was presented for clustering sparse datasets with high dimensions. The algorithm reduces the dispersion within the cluster and increases the negative weight entropy.

Existing solutions for handling mixed data certainly enhanced the capability of the k-means; however, as the solutions do not address the initialization problem, several issues with k-means still remain. Among many proposed measures [67–69], the overlap measure is considered to be the easiest and most computationally efficient method to find similarity between two categorical attributes [70]. One of the shortcomings of this method is inability to differentiate between the distinct attribute values, however, focuses on whether two categorical attributes attain equal values or not. However, after a thorough experimental analysis in [71] on a number of datasets, the authors came to the conclusion that “No single measure is always superior or inferior. This is to be expected since each dataset has different characteristics”. Hence, the overlap measure (for categorical attributes) and Euclidean distance (for numerical attributes) can be combined into a mixed distance measure, as also used in [6,72].

4. Performance Evaluation of k-means Representative Variants

We performed an experimental analysis to investigate different versions of the k-means algorithms for different datasets, especially for the initialization and mixed-data-type problems (GitHub Link: shorturl.at/CR245). All datasets are available in the UCI machine learning repository [73]. Table 3 briefly summarizes the datasets. The next subsections discuss the evaluation metrics used, results analysis and, last but not least, the analysis of computational complexity.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cleveland Heart Disease</td>
<td>Widely used by machine learning researchers. The goal is to detect the presence of heart disease in a patient.</td>
</tr>
<tr>
<td>KDD-Cup 1999 (10%)</td>
<td>Contains standard network traffic that contains different types of cyber attacks simulated in a military network.</td>
</tr>
<tr>
<td>Wisconsin Diagnostic Breast Cancer</td>
<td>Includes features calculated from the images of fine needle aspirate of breast mass.</td>
</tr>
<tr>
<td>Epileptic Seizure Recognition</td>
<td>Commonly used for feature epileptic seizure prediction.</td>
</tr>
<tr>
<td>Credit Approval</td>
<td>Contains a mix of attributes, which makes it interesting to be used with k-means for mixed attributes.</td>
</tr>
<tr>
<td>Postoperative</td>
<td>Contains both categorical and integer values. The missing values are replaced with an average.</td>
</tr>
</tbody>
</table>

4.1. Metrics Used for Experimental Analysis

To evaluate the performance of different algorithms, the following metrics [1–3] were chosen:

- **Accuracy**: This measure outlines the extent to which the predicted labels are in agreement with the true labels. The predicted labels correspond to the class labels where new instances are clustered. The accuracy is calculated by Equation (1).

  \[
  \text{Accuracy} = \frac{\text{Correctly identified class}}{\text{Total number of class}} \times 100
  \]

- **Adjusted rand index (ARI)**: Provides a score of similarity between two different clustering results of the same dataset. For a given set S consisting of a elements and r subsets and two partitions \( Y = \{Y_1, Y_2, \ldots, Y_b\} \) and \( X = \{X_1, X_2, \ldots, X_c\} \), the overlap between the two partitions can be summarized as follows:
The adjusted rand index (ARI) is then calculated by Equation (2):

$$ARI = \frac{\sum_{ij} \binom{Y_i}{2} - \left[\sum_i \binom{r_i}{2} \sum_j \binom{s_j}{2}\right] / \binom{N}{2}}{\frac{1}{2} \left[\sum_i \binom{r_i}{2} + \sum_j \binom{s_j}{2}\right] - \left[\sum_i \binom{r_i}{2} \sum_j \binom{s_j}{2}\right] / \binom{N}{2}}$$}

The ARI score is adjusted to have values between 0 and 1 to represent scores for random and perfect clustering, respectively.

### 4.2. Results

Experimental analysis with \textit{k-means} [9], \textit{x-means} [49], constrained \textit{k-means} [54], \textit{k-prototype} [63], and kernel \textit{k-means} [74] was performed, and a performance comparison in terms of the metrics described above was made. For each of the experiments, a 10 fold cross validation scheme was used to split the dataset, and the reported results consist of average and standard deviation of the scores of these metrics across 10 validation folds. Table 4 addresses the algorithms that deal with initialization issues, while Table 5 encompasses mixed data type oriented methods.

**Table 4.** Comparison of different algorithms addressing initialization issue.

<table>
<thead>
<tr>
<th>Metric</th>
<th>\textit{k-means}</th>
<th>Constrained \textit{k-means}</th>
<th>\textit{x-means}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wisconsin Diagnostic Breast Cancer</td>
<td>0.223 ± 0.310</td>
<td>0.596 ± 0.406</td>
<td>0.086 ± 0.042</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.690 ± 0.134</td>
<td>0.682 ± 0.13</td>
<td>0.683 ± 0.128</td>
</tr>
<tr>
<td>ARI</td>
<td>0.004 ± 0.005</td>
<td>0.107 ± 0.059</td>
<td>0.085 ± 0.169</td>
</tr>
<tr>
<td>KDD Cup 1999</td>
<td>0.195 ± 0.077</td>
<td>0.118 ± 0.087</td>
<td>0.043 ± 0.034</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.004 ± 0.007</td>
<td>0.107 ± 0.059</td>
<td>0.085 ± 0.169</td>
</tr>
<tr>
<td>ARI</td>
<td>0.005 ± 0.001</td>
<td>0.002 ± 0.001</td>
<td>0.002 ± 0.002</td>
</tr>
<tr>
<td>Epileptic Seizure</td>
<td>0.101 ± 0.060</td>
<td>0.099 ± 0.012</td>
<td>0.102 ± 0.053</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.005 ± 0.001</td>
<td>0.002 ± 0.001</td>
<td>0.002 ± 0.002</td>
</tr>
</tbody>
</table>

**Table 5.** Comparison of \textit{k-prototype} and Kernel-\textit{k-means} algorithm datasets with mixed data types.

<table>
<thead>
<tr>
<th>Metric</th>
<th>\textit{k-prototype}</th>
<th>Kernel-\textit{k-means}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit Approval</td>
<td>0.456 ± 0.061</td>
<td>0.437 ± 0.283</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.004 ± 0.005</td>
<td>0.044 ± 0.092</td>
</tr>
<tr>
<td>Cleveland Heart Disease</td>
<td>0.462 ± 0.043</td>
<td>0.590 ± 0.080</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.003 ± 0.001</td>
<td>0.017 ± 0.041</td>
</tr>
<tr>
<td>Post Operative</td>
<td>0.462 ± 0.043</td>
<td>0.590 ± 0.080</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.003 ± 0.001</td>
<td>0.017 ± 0.041</td>
</tr>
</tbody>
</table>
Table 4 summarizes the mean and the standard deviation across five validation folds and compares the performance between regular \( k\text{-means} \), \( x\text{-means} \), and constrained \( k\text{-means} \) algorithms in terms of different metrics for the Wisconsin Diagnostic Breast cancer, KDD Cup 1999 (10\%) and Epileptic Seizure datasets. In Figure 3, it is shown that the algorithm performance varied with different datasets. For example, the \( k\text{-means} \) performed best with the KDD Cup datasets and \( x\text{-means} \) had the worst accuracy. However, the \( x\text{-means} \) performed better than the other two in the Epileptic dataset. The constrained-\( k\text{-means} \) performed best in the Wisconsin dataset. In terms of ARI score, the constrained-\( k\text{-means} \) seemed to perform in a consistent manner. The key takeaway from these results is that there is no algorithm that will provide a consistent solution regardless of the dataset.

![Figure 3. Performance of \( k\text{-means} \) variants addressing initialization issue.](image1)

Table 5 compares the performance between the \( k\text{-prototype} \) and kernel-\( k\text{-means} \) algorithms. These algorithms are suitable for datasets that consist of mixed attributes. In Figure 4, it is reflected that the \( k\text{-prototype} \) performed better using the Credit Approval and Post Operative datasets when considering the accuracy of clustering. However, kernel-\( k\text{-means} \) performed best using the Cleveland Heart Disease dataset. In terms of ARI score comparison, the kernel-\( k\text{-means} \) algorithm consistently performed better than \( k\text{-prototype} \) using all three datasets containing mixed data.

![Figure 4. Performance of \( k\text{-means} \) variants addressing mixed data problem.](image2)

4.3. Computational Complexity Analysis

In addition to the comparison of these variants of \( k\text{-means} \) algorithms in terms of different metrics on different datasets, a time and space complexity comparison of these algorithms was also carried out. Both the time and space complexities of these algorithms depend on the size \( n \) of the datasets. Finding an optimal solution for the \( k\text{-means} \) algorithm is hard in the Euclidean space for both the binary and multi-class clustering problems. The regular \( k\text{-means} \) algorithm have a time complexity
of $O(n^2)$, where $n$ is the size of the input data. However, it can be optimized to be linear and be in the order of $O(n)$ using certain heuristics mentioned in [75,76]. By contrast, the time complexity for constrained-$k$-means is of the order $O(kn)$, where $k$ represents the amount of clusters. It reflects that the constrained-$k$-means has a lower time complexity than the regular $k$-means when the dataset size grows. The $x$-means algorithm, on the other hand, was mainly proposed in order to address the scalability issue of the regular $k$-means algorithm for large datasets, since $x$-means involves a progressive increase in the number of clusters within a user-supplied range ($k_{\text{min}}, k_{\text{max}}$). The time complexity of the $x$-means algorithm is therefore in the order of $O(n \log k_{\text{max}})$. $k_{\text{max}}$ is the maximum number clusters possible in a dataset. The space complexity for all these $k$-means variants is $O((n+k)d)$, where $d$ is the number of features in a dataset. Table 6 shows the complexities of the different clustering algorithms.

<table>
<thead>
<tr>
<th>Complexity</th>
<th>$k$-means</th>
<th>Constrained-$k$-means</th>
<th>$x$-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$O(n^2)$</td>
<td>$O(kn)$</td>
<td>$O(n \log k_{\text{max}})$</td>
</tr>
<tr>
<td>Space</td>
<td>$O((n+k)d)$</td>
<td>$O((n+k)d)$</td>
<td>$O((n+k)d)$</td>
</tr>
</tbody>
</table>

5. Conclusions

In a wide range of application domains, data analysis tasks heavily rely on clustering. This paper focused on the popular $k$-means algorithm and the issues of initialization and inability to handle data with mixed types of features. Unlike other review or survey papers, this paper contains both a critical analysis of the existing literature and an experimental analysis on half a dozen benchmark datasets to demonstrate the performances of different variants of $k$-means. The experimental analysis divulged that there is no universal solution for the problems of the $k$-means algorithm; rather each of the existing variants of the algorithm is either application-specific or data-specific. Our future research will focus on developing a robust $k$-means algorithm that can address both problems simultaneously. This paper will also help the data mining research community to design and develop newer types of clustering algorithms that can address the research issues around Big Data [34,72].

Author Contributions: Conceptualization, M.A.; Data curation, R.S. and M.A.; Formal analysis, M.A., R.S., and S.M.S.I.; Funding acquisition, M.A., R.S.; Investigation, M.A., R.S., and S.M.S.I.; Methodology, M.A.; Project administration, M.A. and S.M.S.I.; Supervision, M.A. and S.M.S.I.; Validation, R.S. and M.A.; Visualization, M.A. and R.S.; and Writing—original draft, M.A. and R.S.; Writing—review and editing, M.A., R.S. and S.M.S.I. All authors have read and agreed to the published version of the manuscript. Authorship must be limited to those who have contributed substantially to the work reported.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References


60. Jiacai, W.; Ruijun, G. An extended fuzzy k-means algorithm for clustering categorical valued data. In Proceedings of the 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI), Sanya, China, 23–24 October 2010; Volume 2, pp. 504–507.

61. Ahmad, A.; Dey, L. A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. 2007, 63, 503–527. [CrossRef]


63. Huang, Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. 1998, 2, 283–304. [CrossRef]


