

8-1-2020

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

Mohiuddin Ahmed

Edith Cowan University, mohiuddin.ahmed@ecu.edu.au

Raihan Seraj

Syed Mohammed Shamsul Islam

Edith Cowan University, syed.islam@ecu.edu.au

Follow this and additional works at: <https://ro.ecu.edu.au/ecuworkspost2013>



Part of the [Computer Sciences Commons](#)

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



Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means Algorithm: A Comprehensive Survey and Performance Evaluation. *Electronics*, 9(8), 1295. <https://doi.org/10.3390/electronics9081295>

This Journal Article is posted at Research Online.

<https://ro.ecu.edu.au/ecuworkspost2013/8567>

Review

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 *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 [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 [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 [8]. Table 1 shows the recent applications on *k-means* clustering [9] in different application domains.

Table 1. Applications of variants of the *k-means* algorithm in different application domains.

Reference	Application	Algorithm
[10]	Face detection	Symmetry-based version of <i>k-means</i> (SBKM).
[11]	Mobile storage positioning	Potential <i>k-means</i> .
[12]	Load pattern	Hierarchical <i>k-means</i> (H-Kmeans).
[13]	Wireless sensor networks	Distributed <i>k-means</i> and fuzzy <i>c-means</i> .
[14]	Partial multiview data	Weighted <i>k-means</i> .
[15]	Mobile health	<i>k-means</i> implemented with CORDIC.
[16]	Endpoint detection	<i>k-means</i> for realtime detection.
[17]	Big data	Privacy preserving <i>k-means</i> .
[18]	Multiview data	<i>k-means</i> .
[19]	Wind power forecasting	<i>k-means</i> with bagging neural network.
[20]	Social tags	<i>k-means</i> based on latent semantic analysis.
[21]	Sensing for IGBT current	<i>k-means</i> with neural network.
[22]	Image segmentation.	Kernel <i>k-means</i> Nystrom approximation.
[23]	Image compression	<i>k-means</i> cuckoo optimization.
[24]	Sound source angle estimation.	Neural network based on global <i>k-means</i> .
[25]	Shape recognition	Fuzzy <i>k-means</i> clustering ensemble (FKMCE).
[26]	Signal processing	Compressive <i>k-means</i> clustering (CKM).
[27]	Text processing	Vanilla <i>k-means</i> .
[28]	High dimensional data processing	Fast adaptive <i>k-means</i> (FAKM).
[29]	Computational complexity	Multiple kernel <i>k-means</i> with late fusion.
[30]	Image processing	A hybrid parallelization of <i>k-means</i> algorithm.
[31]	Adaptive clustering	Fuzzy <i>k-means</i> with S-distance.
[32]	DDoS detection	Semi-supervised <i>k-means</i> algorithm with hybrid feature.
[33]	Optimization	Non alternating stochastic <i>k-means</i> .
[34]	Data Summarization	Modified <i>x-means</i> .

This survey studies the problems of and solutions to partition-based clustering, and more specifically the widely used *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, *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 *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 *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 *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 *k-means* algorithms to overcome these impediments, they are however domain specific and do not generalize well. For example, a *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 *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 *k-means* algorithm.
- Experiments are performed using the improved *k-means* algorithms to find out their effectiveness using six benchmark datasets.

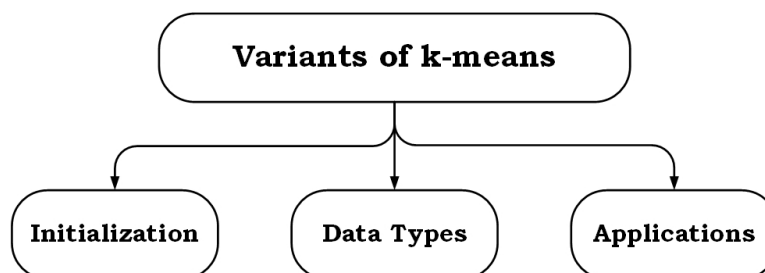


Figure 1. A simple taxonomy of variants of *k-means* algorithm.

Table 2. Comparison with existing surveys.

Survey	Initialization	Data Types	Applications	Experiments
Yang [38]	✓	×	×	×
Filippone [39]	✓	×	×	×
Rai [40]	✓	×	×	×
This paper	✓	✓	✓	✓

Paper Roadmap

Our paper is structured as follows. Sections 2 and 3 address the problems of the *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 *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*.

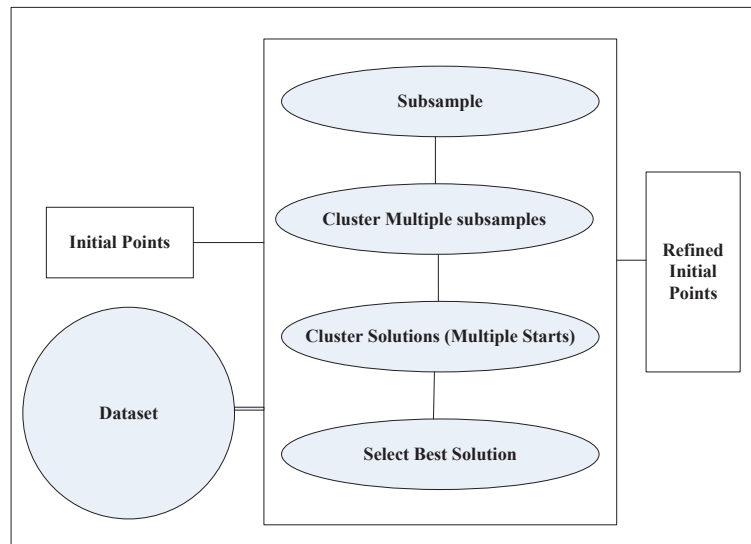


Figure 2. Refined initial instances, adapted from [54].

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 3. Summary of datasets.

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

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).

$$Accuracy = \frac{\text{Correctly identified class}}{\text{Total number of class}} \times 100 \quad (1)$$

- **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 α elements and r subsets and two partitions $Y = \{Y_1, Y_2, \dots, Y_b\}$ and $X = \{X_1, X_2, \dots, X_c\}$, the overlap between the two partitions can be summarized as follows:

	Y_1	Y_2	...	Y_b	Sums
X_1	α_{11}	α_{12}		α_{1b}	r_1
X_2	α_{21}	α_{22}		α_{2b}	r_2
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
X_c	α_{c1}	α_{c2}	...	α_{cb}	r_c
Sums	s_1	s_2	...	s_b	

The adjusted rand index (ARI) is then calculated by Equation (2):

$$ARI = \frac{\sum_{ij} \binom{\alpha_{ij}}{2} - [\sum_i \binom{r_i}{2} \sum_j \binom{s_j}{2}] / \binom{\alpha}{2}}{\frac{1}{2} [\sum_i \binom{r_i}{2} + \sum_j \binom{s_j}{2}] - [\sum_i \binom{r_i}{2} \sum_j \binom{s_j}{2}] / \binom{\alpha}{2}} \tag{2}$$

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 *k-means* [9], *x-means* [49], constrained *k-means* [54], *k-prototype* [63], and kernel *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.

Metric	<i>k-means</i>	Constrained <i>k-means</i>	<i>x-means</i>
Wisconsin Diagnostic Breast Cancer			
Accuracy	0.223 ± 0.310	0.596 ± 0.406	0.086 ± 0.042
ARI	0.690 ± 0.134	0.682 ± 0.13	0.683 ± 0.128
KDD Cup 1999			
Accuracy	0.195 ± 0.077	0.118 ± 0.087	0.045 ± 0.034
ARI	0.004 ± 0.007	0.107 ± 0.059	0.085 ± 0.169
Epileptic Seizure			
Accuracy	0.101 ± 0.060	0.099 ± 0.012	0.102 ± 0.053
ARI	0.005 ± 0.001	0.002 ± 0.001	0.002 ± 0.002

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

Metric	<i>k-prototype</i>	Kernel- <i>k-means</i>
Credit Approval		
Accuracy	0.456 ± 0.061	0.437 ± 0.283
ARI	0.004 ± 0.005	0.044 ± 0.092
Cleveland Heart Disease		
Accuracy	0.462 ± 0.043	0.590 ± 0.080
ARI	0.003 ± 0.001	0.017 ± 0.041
Post Operative		
Accuracy	0.462 ± 0.043	0.590 ± 0.080
ARI	0.003 ± 0.001	0.017 ± 0.041

Table 4 summarizes the mean and the standard deviation across five validation folds and compares the performance between regular *k-means*, *x-means*, and constrained *k-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-means* performed best with the KDD Cup datasets and *x-means* had the worst accuracy. However, the *x-means* performed better than the other two in the Epileptic dataset. The constrained-*k-means* performed best in the Wisconsin dataset. In terms of ARI score, the constrained-*k-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.

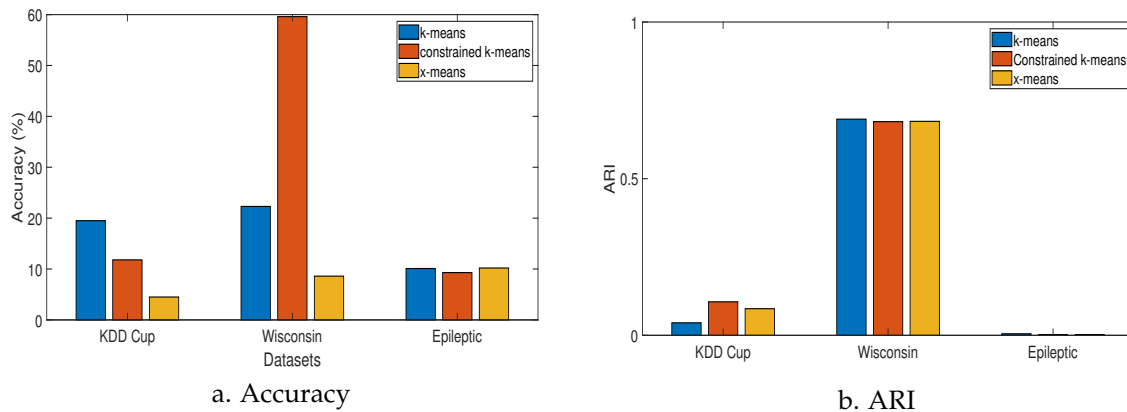


Figure 3. Performance of *k-means* variants addressing initialization issue.

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

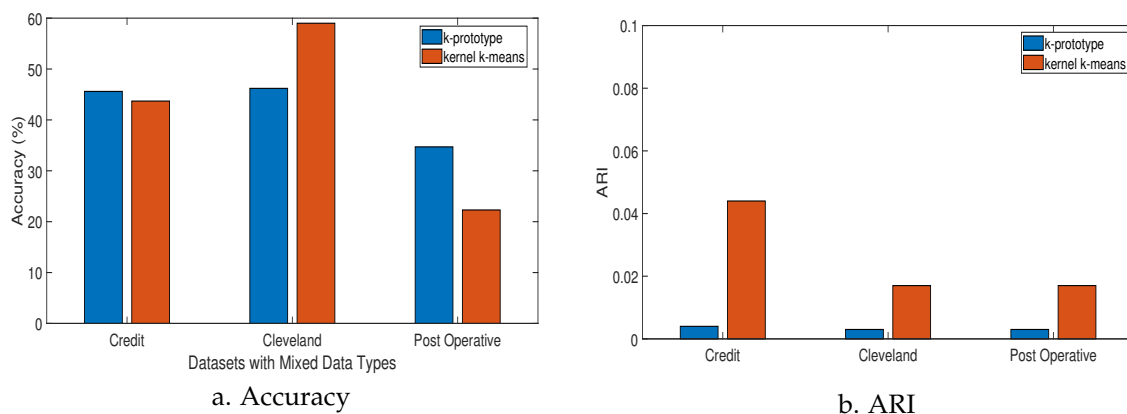


Figure 4. Performance of *k-means* variants addressing mixed data problem.

4.3. Computational Complexity Analysis

In addition to the comparison of these variants of *k-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-means* algorithm is hard in the Euclidean space for both the binary and multi-class clustering problems. The regular *k-means* algorithm have a time complexity

of $\mathcal{O}(n^2)$, where n is the size of the input data. However, it can be optimized to be linear and be in the order of $\mathcal{O}(n)$ using certain heuristics mentioned in [75,76]. By contrast, the time complexity for constrained- k -means is of the order $\mathcal{O}(kn)$, 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_{min}, k_{max}). The time complexity of the x -means algorithm is therefore in the order of $\mathcal{O}(n \log k_{max})$. k_{max} is the maximum number clusters possible in a dataset. The space complexity for all these k -means variants is $\mathcal{O}((n+k)d)$, where d is the number of features in a dataset. Table 6 shows the complexities of the different clustering algorithms.

Table 6. Complexities of different variants of k -means.

Complexity	k -means	Constrained- k -means	x -means
Time	$\mathcal{O}(n^2)$	$\mathcal{O}(kn)$	$\mathcal{O}(n \log k_{max})$
Space	$\mathcal{O}((n+k)d)$	$\mathcal{O}((n+k)d)$	$\mathcal{O}((n+k)d)$

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

1. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2012.
2. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
3. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: A review. *ACM Comput. Surv.* **1999**, *31*, 264–323. [[CrossRef](#)]
4. Ahmed, M.; Choudhury, V.; Uddin, S. Anomaly detection on big data in financial markets. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Sydney, Australia, 31 July–3 August 2017; pp. 998–1001.
5. Ahmed, M. An unsupervised approach of knowledge discovery from big data in social network. *EAI Endorsed Trans. Scalable Inf. Syst.* **2017**, *4*, 9. [[CrossRef](#)]
6. Ahmed, M. Collective anomaly detection techniques for network traffic Analysis. *Ann. Data Sci.* **2018**, *5*, 497–512. [[CrossRef](#)]
7. Tondini, S.; Castellán, C.; Medina, M.A.; Pavesi, L. Automatic initialization methods for photonic components on a silicon-based optical switch. *Appl. Sci.* **2019**, *9*, 1843. [[CrossRef](#)]

8. Ahmed, M.; Mahmood, A.N.; Islam, M.R. A survey of anomaly detection techniques in financial domain *Future Gener. Comput. Syst.* **2016**, *55*, 278–288.
9. MacQueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, USA, 1 January 1967; pp. 281–297.
10. Su, M.C.; Chou, C.H. A modified version of the k-means algorithm with a distance based on cluster symmetry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 674–680.
11. Cabria I.; Gondra, I. Potential-k-means for load balancing and cost minimization in mobile recycling network. *IEEE Syst. J.* **2014**, *11*, 242–249. [[CrossRef](#)]
12. Xu, T.S.; Chiang, H.D.; Liu, G.Y.; Tan, C.W. Hierarchical k-means method for clustering large-scale advanced metering infrastructure data. *IEEE Trans. Power Deliv.* **2015**, *32*, 609–616. [[CrossRef](#)]
13. Qin, J.; Fu, W.; Gao, H.; Zheng, W.X. Distributed k-means algorithm and fuzzy c-means algorithm for sensor networks based on multiagent consensus theory. *IEEE Trans. Cybern.* **2016**, *47*, 772–783. [[CrossRef](#)]
14. Liu, H.; Wu, J.; Liu, T.; Tao, D.; Fu, Y. Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1129–1143. [[CrossRef](#)]
15. Adapa, B.; Biswas, D.; Bhardwaj, S.; Raghuraman, S.; Acharyya, A.; Maharatna, K. Coordinate rotation-based low complexity k-means clustering Architecture. *IEEE Trans. Very Large Scale Integr. Syst.* **2017**, *25*, 1568–1572. [[CrossRef](#)]
16. Jang, H.; Lee, H.; Lee, H.; Kim, C.K.; Chae, H. Sensitivity enhancement of dielectric plasma etching endpoint detection by optical emission spectra with modified k-means cluster analysis. *IEEE Trans. Semicond. Manuf.* **2017**, *30*, 17–22. [[CrossRef](#)]
17. Yuan, J.; Tian, Y. Practical privacy-preserving mapreduce based k-means clustering over large-scale dataset *IEEE Trans. Cloud Comput.* **2017**, *7*, 568–579.
18. Xu, J.; Han, J.; Nie, F.; Li, X. Re-weighted discriminatively embedded k-means for multi-view clustering. *IEEE Trans. Image Process.* **2017**, *26*, 3016–3027. [[CrossRef](#)]
19. Wu, W.; Peng, M. A data mining approach combining k-means clustering with bagging neural network for short-term wind power forecasting. *IEEE Internet Things J.* **2017**, *4*, 979–986. [[CrossRef](#)]
20. Yang, J.; Wang, J. Tag clustering algorithm lmmsk: Improved k-means algorithm based on latent semantic analysis. *J. Syst. Electron.* **2017**, *28*, 374–384.
21. Zeng, X.; Li, Z.; Gao, W.; Ren, M.; Zhang, J.; Li, Z.; Zhang, B. A novel virtual sensing with artificial neural network and k-means clustering for igbt current measuring. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7343–7352. [[CrossRef](#)]
22. He, L.; Zhang, H. Kernel k-means sampling for nyström approximation. *IEEE Trans. Image Process.* **2018**, *27*, 2108–2120. [[CrossRef](#)]
23. Manju, V.N.; Fred, A.L. Ac coefficient and k-means cuckoo optimisation algorithm-based segmentation and compression of compound images. *IET Image Process.* **2017**, *12*, 218–225. [[CrossRef](#)]
24. Yang, X.; Li, Y.; Sun, Y.; Long, T.; Sarkar, T.K. Fast and robust rbf neural network based on global k-means clustering with adaptive selection radius for sound source angle estimation. *IEEE Trans. Antennas Propag.* **2018**, *66*, 3097–3107. [[CrossRef](#)]
25. Bai, L.; Liang, J.; Guo, Y. An ensemble clusterer of multiple fuzzy k-means clusterings to recognize arbitrarily shaped clusters. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 3524–3533. [[CrossRef](#)]
26. Schellekens, V.; Jacques, L. Quantized compressive k-means *IEEE Signal Process. Lett.* **2018**, *25*, 1211–1215.
27. Alhawarat, M.; Hegazi, M. Revisiting k-means and topic modeling, a comparison study to cluster arabic documents. *IEEE Access* **2018**, *6*, 740–749. [[CrossRef](#)]
28. Wang, X.D.; Chen, R.C.; Yan, F.; Zeng, Z.Q.; Hong, C.Q. Fast adaptive k-means subspace clustering for high-dimensional data. *IEEE Access* **2019**, *7*, 639–651. [[CrossRef](#)]
29. Wang, S.; Zhu, E.; Hu, J.; Li, M.; Zhao, K.; Hu, N.; Liu, X. Efficient multiple kernel k-means clustering with late fusion. *IEEE Access* **2019**, *7*, 109–120. [[CrossRef](#)]
30. Kwedlo, W.; Czochanski, P.J. A hybrid mpi/openmp parallelization of k-means algorithms accelerated using the triangle inequality. *IEEE Access* **2019**, *7*, 280–297. [[CrossRef](#)]
31. Karlekar, A.; Seal, A.; Krejcar, O.; Gonzalo-Martin, C. Fuzzy k-means using non-linear s-distance. *IEEE Access* **2019**, *7*, 121–131. [[CrossRef](#)]

32. Gu, Y.; Li, K.; Guo, Z.; Wang, Y. Semi-supervised k-means ddos detection method using hybrid feature selection algorithm. *IEEE Access* **2019**, *7*, 351–365. [[CrossRef](#)]
33. Lee, M. Non-alternating stochastic k-means based on probabilistic representation of solution space. *Electron. Lett.* **2019**, *55*, 605–607. [[CrossRef](#)]
34. Ahmed, M. Data summarization: A survey. *Knowl. Inf. Syst.* **2019**, *58*, 249–273. [[CrossRef](#)]
35. Wu, X.; Kumar, V.; Quinlan, J.R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Philip, S.Y.; et al. Top 10 algorithms in data Mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [[CrossRef](#)]
36. Tian, K.; Zhou, S.; Guan, J. Deepcluster: A general clustering framework based on deep learning. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Skopje, Macedonia, 18–22 September 2017; pp. 809–825.
37. He, B.; Qiao, F.; Chen, W.; Wen, Y. Fully convolution neural network combined with k-means clustering algorithm for image segmentation. In Proceedings of the Tenth International Conference on Digital Image Processing (ICDIP 2018), Shanghai, China, 11–14 May 2018; Volume 10806, pp. 1–7.
38. Yang, M.S. A survey of fuzzy clustering. *Math. Comput.* **1993**, *18*, 1–16. [[CrossRef](#)]
39. Filippone, M.; Camastra, F.; Masulli, F.; Rovetta, S. A survey of kernel and spectral methods for clustering. *Pattern Recognit.* **2008**, *41*, 176–190. [[CrossRef](#)]
40. Rai, P.; Singh, S. A survey of clustering techniques. *Int. Comput. Appl.* **2010**, *7*, 1–5. [[CrossRef](#)]
41. Yu, H.; Wen, G.; Gan, J.; Zheng, W.; Lei, C. Self-paced learning for k-means clustering algorithm. *Pattern Recognit. Lett.* **2018**. [[CrossRef](#)]
42. Ye, S.; Huang, X.; Teng, Y.; Li, Y. K-means clustering algorithm based on improved cuckoo search algorithm and its application. In Proceedings of the 2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China, 9–12 March 2018; pp. 422–426.
43. Ben-David, S.; Von Luxburg, U.; Pál, D. A sober look at clustering stability. In Proceedings of the International Conference on Computational Learning Theory, San Diego, CA, USA, 13–15 June 2006; pp. 5–19.
44. Bubeck, S.; Meilä, M.; von Luxburg, U. How the initialization affects the stability of the k-means algorithm. *ESAIM Probab. Stat.* **2012**, *16*, 436–452. [[CrossRef](#)]
45. Melnykov, I.; Melnykov, V. On k-means algorithm with the use of mahalanobis Distances. *Stat. Probab. Lett.* **2014**, *84*, 88–95. [[CrossRef](#)]
46. Ball, G.H.; Hall, D.J. A clustering technique for summarizing multivariate data. *Syst. Res. Behav. Sci.* **1967**, *12*, 153–155. [[CrossRef](#)]
47. Carpenter, G.A.; Grossberg, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Comput. Vis. Graph. Image Process.* **1987**, *37*, 54–115. [[CrossRef](#)]
48. Xu, R.; Wunsch, D. *Clustering*; Wiley-IEEE Press: Piscataway, NJ, USA, 2009.
49. Pelleg, D.; Moore, A.W. X-means: Extending k-means with efficient estimation of the number of clusters. In Proceedings of the 17th International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000; Volume 1, pp. 727–734.
50. Bozdogan, H. Model selection and akaike’s information criterion (AIC): The general theory and its analytical extensions. *Psychometrika* **1987**, *52*, 345–370. [[CrossRef](#)]
51. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [[CrossRef](#)]
52. Ahmed, M.; Barkat Ullah, A.S.S.M. Infrequent pattern mining in smart healthcare environment using data summarization. *J. Supercomput.* **2018**, *74*, 5041–5059. [[CrossRef](#)]
53. Ahmed, M.; Mahmood, A. Network traffic analysis based on collective anomaly Detection. In Proceedings of the 9th IEEE International Conference on Industrial Electronics and Applications, Hangzhou, China, 9–11 June 2004; pp. 1141–1146.
54. Bradley, P.S.; Fayyad, U.M. Refining initial points for k-means Clustering. *ICML* **1998**, *98*, 91–99.
55. Pena, J.M.; Lozano, J.A.; Larranaga, P. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognit. Lett.* **1999**, *20*, 1027–1040. [[CrossRef](#)]
56. Forgy, E.W. Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics* **1965**, *21*, 768–769.
57. Kaufman, L.; Rousseeuw, P.J. *Finding Groups in Data: An Introduction to Cluster Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 344.
58. Hussain, S.F.; Haris, M. A k-means based co-clustering (kcc) algorithm for sparse, high dimensional data. *Expert Syst. Appl.* **2019**, *118*, 20–34. [[CrossRef](#)]

59. Gupta, S.; Rao, K.S.; Bhatnagar, V. K-means clustering algorithm for categorical attributes. In Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, Florence, Italy, 30 August–1 September 1999; pp. 203–208.
60. Jiakai, 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](#)]
62. Couto, J. Kernel k-means for categorical data. In *International Symposium on Intelligent Data Analysis*; Springer: Berlin, Germany, 2005; pp. 46–56.
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](#)]
64. Bai, L.; Liang, J.; Dang, C.; Cao, F. The impact of cluster representatives on the convergence of the k-modes type clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1509–1522. [[CrossRef](#)]
65. Dzogang, F.; Marsala, C.; Lesot, M.; Rifqi, M. An ellipsoidal k-means for document clustering. In Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM), Brussels, Belgium, 10–13 December 2012; pp. 221–230.
66. Jing, L.; Ng, M.K.; Huang, J.Z. An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 1026–1041. [[CrossRef](#)]
67. Cramér, H. *The Elements of Probability Theory and Some of Its Applications*; John Wiley & Sons: New York, NY, USA, 1954.
68. Maung, K. Measurement of association in a contingency table with special reference to the pigmentation of hair and eye colours of scottish school children. *Ann. Eugen.* **1941**, *11*, 189–223. [[CrossRef](#)]
69. Pearson, K. On the general theory of multiple contingency with special reference to partial contingency. *Biometrika* **1916**, *11*, 145–158. [[CrossRef](#)]
70. Stanfill, C.; Waltz, D. Toward memory-based reasoning. *Commun. ACM* **1986**, *29*, 1213–1228. [[CrossRef](#)]
71. Boriah, S.; Chandola, V.; Kumar, V. Similarity measures for categorical data: A comparative evaluation. In Proceedings of the SIAM International Conference on Data Mining, Atlanta, GA, USA, 24–26 April 2008; pp. 243–254.
72. Ahmed, M. Detecting Rare and Collective Anomalies in Network Traffic Data Using Summarization. 2016. Available online: <http://handle.unsw.edu.au/1959.4/56990> (accessed on 29 May 2020).
73. Dheeru, D.; Karra Taniskidou, E. UCI Machine Learning Repository. 2017. Available online: <http://archive.ics.uci.edu/ml> (accessed on 29 May 2020).
74. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering Algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [[CrossRef](#)]
75. Pakhira, M.K. A linear time-complexity k-means algorithm using cluster Shifting. In Proceedings of the 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 14–16 November 2014; pp. 1047–1051.
76. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and Implementation. *IEEE Trans. Pattern Anal. Mach.* **2002**, *7*, 881–892. [[CrossRef](#)]

