ISSN:0975-9646

Improving the *k*-means Algorithm by Data-Censoring and MaxMin Center Selection

James M. Phillips[#1], Suk J. Seo[*2]

[#]*Computer Science, Middle Tennessee State University, Murfreesboro, Tennessee, United States*

1jmp9q@mtmail.mtsu.edu

2suk.seo@mtsu.edu

***Abstract*— The *k*-means algorithm is one of the most widely used clustering algorithms in many different fields such as the diagnosis of cancers and species classification. Despite its popularity, *k*-means has some challenges in terms of run time and accuracy. While the algorithm can get reasonable accuracy in clustering, it is heavily reliant on the number of clusters and the selection of initial starting points of the centers. Additionally, the data needs to be in its most usable state, that is, the data must have labels for testing, well-distributed within each classification containing few outliers. We have addressed these issues by using data-censoring techniques and a variation of the MaxMin algorithm for initial center selection. The results of our experimentation have shown that the new approach has reduced overall runtime while maintaining similar accuracy or better.**

***Keywords*— Clustering, *k*-means clustering algorithm, data mining, data-censoring, MaxMin algorithm**

I. INTRODUCTION

A. The k-means Algorithm

In 1967, J. MacQueen [1] created the *k*-means clustering algorithm, which he believed would primarily be used for "similarity grouping." In the *k*-means algorithm, the number *k* represents the desired number of clusters, which is provided by the user. Given a set of data points and a number *k*, the algorithm generates randomly the initial center (or centroid) of each of *k* clusters, as described using pseudo-code in Algorithm 1. Then, the algorithm computes the distance from each data point to each of the *k* centers and assigns the data point to the cluster whose center is closest using the Euclidean Distance (ED) between two points *p* and *q*, denoted as *d*(*p*, *q*) where *i* through *n* represents the columns in each data point as shown below.

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

Next, the centroids are updated by taking the means of the data points within the same group. With the updated centroids, the algorithm computes the distance from each data point and reassigns the data point to a different group if necessary. The process repeats until no data points change their group. This provides a learned relationship pattern amongst data points. However, he never intended for the algorithm to be used for exact groupings. The goal of the *k*-means algorithm was to create a reasonably good grouping that would help the user understand abundant amounts of N-dimensional data.

Algorithm 1: Base *k*-means algorithm

Input:
- Set of n data points $\{d_1, d_2, ..., d_n\}$
- Number of desired clusters, k

Output:
- Set of k clusters, $C = \{C_1, C_2, ..., C_k\}$

Steps:
1) For each cluster C_j, $1 \leq i \leq k$, select a data point at random as the initial centroid of C_j.
2) For each data point d_i, $1 \leq i \leq n$, assign it to a cluster.
   - For each cluster C_j, compute the ED from the centroid of C_j to d_i.
   - Assign d_i to the cluster with the lowest ED.
3) For each cluster C_j, $1 \leq i \leq k$, update the centroid
   - Take the average of all the data points in C_j and let the average be the new centroid of C_j
4) Repeat Steps 2) and 3) until
   - No data points move to a different cluster in Step 3).

B. Why is k-means Important?

The *k*-means algorithm is an unsupervised learning technique, that is, it makes its correlations between the different properties of an entry rather than basing its conclusions on label selection. It is also considered a non-parametric method. Non-parametric methods are primarily used when functional relationships between attributes can potentially be very difficult to predict.

With *k*-means clustering, we assume that the data can be represented using a small subset of data points which express the average or mean behavior of the data at each point (shown in Figure 1). With the growing world of Big Data, being able to reduce data sets that contain Gigabytes to Terabytes worth of data is becoming more of a necessity than ever before and this data reduction is where *k*-means excels at.

C. The Problems with k-means

The *k*-means algorithm is a very popular clustering method, but it has some drawbacks. For example, the algorithm requires the user to provide the value of *k*, which is the number of clusters they want to use. Furthermore, the original intent for the algorithm was for the user to run several times with different starting values

and to see any skews in the data that could be lowering the accuracy of the algorithm [1]. Another reason for necessitating multiple runs is that the algorithm has a good chance of showing poor convergence of the clusters because the initial centroids of clusters are placed randomly. Some methods start at a single point in the data set while others simply plot a point within the bounds of the data. Because of this randomness, the clusters can run into issues of having too few data samples or even having no data points at all.

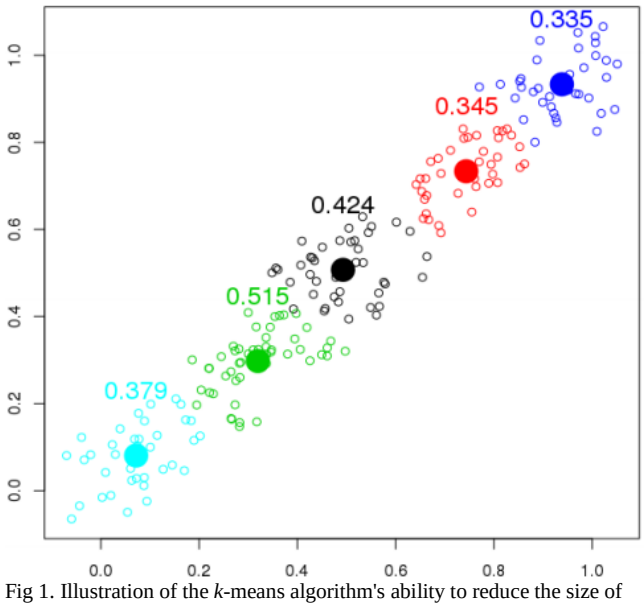

Fig 1. Illustration of the *k*-means algorithm's ability to reduce the size of the data in points centered in groupings.

In the next section, we discuss background information that is used in developing our new approach. In Section 3, we describe our proposed improved *k*-means algorithm. Section 4 explains the necessary components needed to recreate this project from scratch and describes our testing process. Section 5 shows and compares the test results of the base *k*-means algorithm and the Improved *k*-means algorithm. We conclude with Section 6 discussing the implication of our results and future paths this project can take.

II. Background

A. Early Projects

Many researchers have tried to find a way of choosing better initial starting centers, for the *k*-means algorithm, to converge to the global optimum accuracy. For example, in 1998 Paul Bradley and Usama Fayyad introduced a way of refining the initial starting points by “operating over small sub-samples of a given database [2]." This allows their algorithm to operate around the modes (maxima) of the data set. By operating around the modes, they can reduce the total number of iterations the *k*-means algorithm must go through before convergence is achieved.

Bradly [3] later tried to reduce the number of empty clusters generated by the original *k*-means algorithm. His proposed method forces every cluster to contain a certain amount of points by using a constraint method to gather outlier data that does not fit within the normal clusters of the data.

Other approaches focused on finding starting points rather than controlling the data points. For example, Ting Su and Jennifer Dy used a deterministic approach to find the initial starting points. Their algorithm splits the sample data hierarchically by dividing the cluster into halves. This process is continued across the cluster with the highest sum-squared-error until *k* clusters are found allowing for a more balanced set of data points [4].

By balancing out the clusters, they tried to make the algorithm produce much more difficult to have bad clusters, which contain few or no data points. Bad clusters with few data points are generated when those few data points share some very specific characteristics, and this prevents them from fitting in a larger cluster. A cluster generated with no data points suggests that the cluster's initial starting location is likely too close to another and hence it becomes essentially useless.

B. Initialization Techniques

Madhu Yedla et al. suggest that fixing attributes to be non-negative and picking initial centroids based on the distance from the origin (see [5] for the detail.). After finding all the distances, the algorithm splits the data into *k* groups based on the distance measurements and picks the mean of each of those sets as a centroid [5]. Other projects, like Pasi Fränti and Sami Sieranoja's, continued Yedla's work but used a different heuristic-based distance approach to determine distance from the origin and checked the results of accuracy when repeating this calculation multiple times [6].

In this paper, we propose a new approach that utilizes two methods. One is the application of a variation of the MaxMin algorithm for the initial centroid selection of the *k* clusters that would improve the efficiency of the k-mean algorithm by allowing the centroid adjustment only once instead of multiple iterations. The other is a data-censoring technique, which removes outliers from the data set. These two methods will be described in detail in the next two sections.

III. The Improved *k*-Means Algorithm

We have modified the *k*-means algorithm to increase efficiency while maintaining similar accuracy measures. Algorithm 2 shows the pseudo-code of the Improved *k*-means Algorithm. Instead of randomly selecting the initial location of all the *k* clusters, the Improved *k*-means algorithm chooses a data point randomly for the initial center of only the first cluster. Next, for the second cluster, it selects the point that is the farthest away from the first center using ED. In determining the initial center of all the remaining (*k*-2) clusters, the algorithm uses a variation of the MaxMin algorithm as described below.

First, for each data point, the algorithm computes the ED measurement from each of the centers of the existing clusters. Next, it computes the sum of the EDs for each data point. And then, using the 10 data points with the highest distance sum, the algorithm computes the standard deviation. Finally, it selects the point with the lowest

standard deviation as the center of the next cluster. This process is repeated until all the *k* centers have been selected. This extra step of finding the standard deviation is needed to find the point that is the farthest from all the previously selected centers. Without using the standard deviation, there could be a case where the newly selected center is very close to an existing center, but it has been chosen because it is very far from another center. By taking the highest EDs and then the lowest standard deviation we look for the value with the lowest difference in distance measurements from each center.

Once the centers of all of the *k* clusters are selected, the algorithm assigns each data point to its closest cluster, and then it adjusts the value of the centroid of each cluster as the final step by taking the average of all the data points that belong to the same cluster. Unlike the base *k*-means algorithm, our algorithm does not repeat the step of adjusting the centers; it is done only once. Our Improved *k*-means algorithm takes longer in selecting the initial centers of the *k* clusters, but it compensates the time spent on initial center selection by saving a significant amount of time in updating the value of centroids.

Algorithm 2: Improved *k*-means algorithm

Input:
Set of n data points $\{d_1, d_2, ..., d_n\}$
Number of desired clusters, k

Output:
Set of k clusters, $C = \{C_1, C_2, ..., C_k\}$

Steps:
1) For each cluster C_j, $1 \le i \le k$, select its initial centroid.
a) Select a data point at random as the centroid of the first cluster, C_1.
b) Determine the centroid of C_2.
i) For d_i, $1 \le i \le n$, compute the ED to the centroid of C_1.
ii) Select the data point with the maximum distance as the centroid of C_2.
c) For C_j, $3 \le i \le k$, select its centroid.
i) For each data point d_i, $1 \le i \le n$, compute the ED from each of the centroids of C_m, $3 \le m \le j$-1.
ii) For each data point d_i, $1 \le i \le n$, compute the sum of the EDs obtained in Step i).
iii) Take the top 10 data points with the largest sum of the EDs.
iv) For each of the 10 data points, compute the average of the ED sum and the standard deviation.
v) Select the data point with the lowest standard deviation as the centroid of C_j.
2) For each data point d_i, $1 \le i \le n$, assign it to a cluster.
a) For each C_j, compute the ED from the centroid of C_j to d_i.
b) Assign d_i to the cluster with the lowest ED.
3) For each cluster C_j, $1 \le i \le k$, update the centroid
Take the average of all the data points in C_j and let the average be the new centroid of C_j

IV. METHODS

A. Data Sets, Similarities, and Center Selection

For our experimentation, we used the Iris and Breast Tissue data sets provided by [7]. Each data point in the Iris data set contains four attributes represented by columns. Each data point belongs to one of the three classes, labelled as 0, 1, and 2, and these labels appear in the final column of the data set. Some of the main reasons we have chosen the Iris data set are that it is well balanced with each of the labels containing exactly 50 samples, that data contained in each column is within a small distance from each other data point, and that no data point is missing any column information. The Breast Tissue data set was chosen because of its stark difference from the Iris data set. The Breast Tissue data set contains six classes, represented by the values 0, 1, 2, 3, 4, and 5, contained in the final column of the data set. The data set has nine attributes that can have a large range of values and there is an unbalanced amount of data samples for each class with the most frequent class being class 5 at 22 instances and the least frequent being class 4 with 14 instances.

For the experiment, we have used the original versions of the two data sets as well as a data-censored version of each data set. As mentioned previously in Section 2, the use of data censoring and outlier collection has been proven useful in increasing the accuracy of the *k*-means algorithm. We have used a variation of both these methods against our data sets by zero centering our data per column and finding the Z-score of each attribute in the data set. Algorithm 3 shows the data-censoring procedure in pseudo-code. Note that Step 1) in Algorithm 3 describes how to determine the Z-score for each of the column/attribute values. By taking the Z-score we can find the data points along with different attributes that would be considered outlier data amongst the current set of data points. The data points that yielded a Z-score with a magnitude greater than 3.0 were then removed from the data set.

Algorithm 3: Data-censoring

Input:
Set of n data points $\{d_1, d_2, ..., d_n\}$

Output:
Set of m data points $\{d_1, d_2, ..., d_m\}$, $m \le n$.

Steps:
1) For each data point d_i, determine the Z-Score for each of the column/attribute values.
a) Compute the average of the column value of all the data points, μ, and the standard deviation, σ.
b) Calculate the Z-score for the column value, x, of each data point by the formula: $z = (x - \mu)/\sigma$
2) Remove data points having at least one column with a Z-score higher than 3.
3) Process each column so that the mean of the column is 0.
a) Find the mean of the column.
b) Subtract the mean from the respective column of each data point

In our experiment, both the *k*-means and the Improved *k*-means algorithms take in a training set, a testing set, and the number of clusters. The testing set is a selection of 10 random data points while the training set consists of all the remaining data points. The training and testing sets were generated randomly from the available data points and no points appear in both the training and testing set.

To decide on the choice of *k*, each number between 1 and (n-10), where n is equal to the number of data points in the data set, was tested 100 times as a candidate for the value of *k*. Figures 2, 3, 4, and 5 below show the results of the tests conducted on the four data sets: the original and data-censored versions of the Iris and the Breast Tissue data sets, respectively. As suggested in Figures 2 and 3, the tests lead to using 20 for the value of *k* for both versions of the Iris data set. Similarly, Figures 4 and 5 showing the test results on the Breast Tissue data set suggest that 90 is the optimal value for *k*.

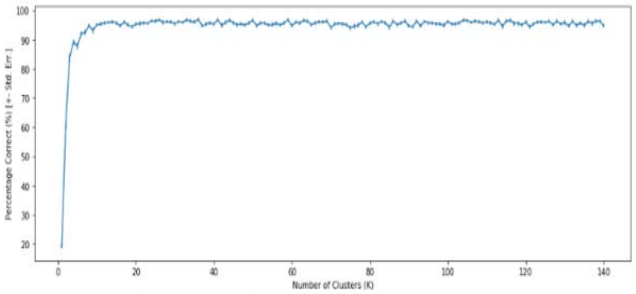

Fig 2. Results from the test for the value of *k* using the original Iris data set. The y-axis reads Percentage Correct (% | +- Std Err) and has a scale of 0-100. The x-axis reads Number of Clusters (*k*) and has a scale of 0-140.

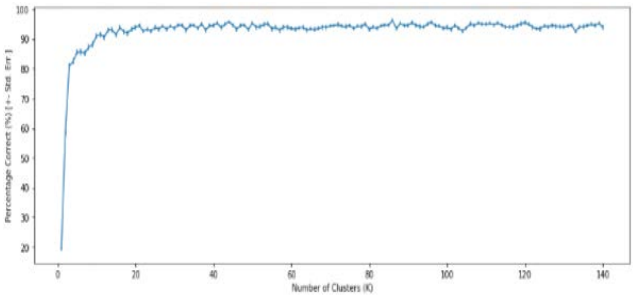

Fig 3. Results from the test for the value of *k* using the censored Iris data set. The y-axis reads Percentage Correct (% | +- Std Err) and has a scale of 0-100. The x-axis reads Number of Clusters (*k*) and has a scale of 0-140.

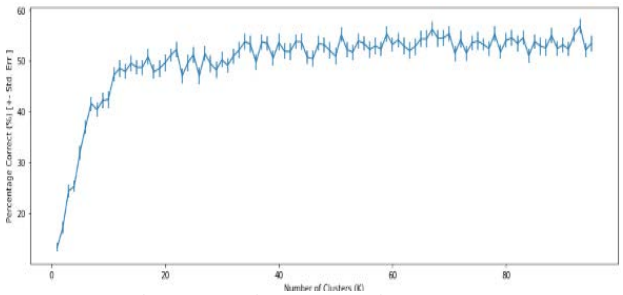

Fig 4. Results from the test for the value of *k* using the original Breast Tissue data set. The y-axis reads Percentage Correct (% | +- Std Err) and has a scale of 0-60. The x-axis reads Number of Clusters (*k*) and has a scale of 0-95.

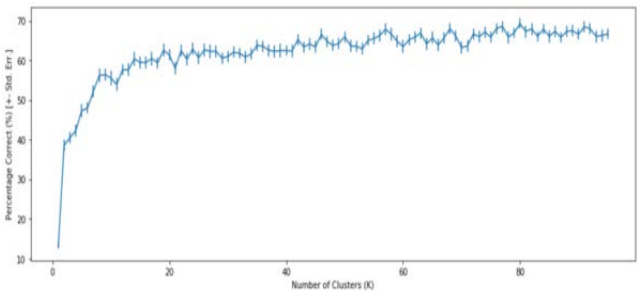

Fig 5. Results from the test for the value of *k* using the censored Breast Tissue data set. The y-axis reads Percentage Correct (% | +- Std Err) and has a scale of 0-70. The x-axis reads Number of Clusters (*k*) and has a scale of 0-95.

B. Experimentation

To compare the accuracy of the two algorithms, we have implemented both algorithms in Python and tested the two programs using the four data sets described above. The programs used 20 and 90 as the value of *k* for the Iris and the Breast Tissue data sets, respectively. At each run the programs randomly selected 10 data points as the testing set (and the remaining (n-10) data points as the training set) and recorded the number of classifications it produced correctly, ranging from 0-10. The accuracy was estimated by taking the average of 100 runs for each program on each data set.

For the comparison of the efficiency (time complexity) of the two algorithms, the programs also calculated the time they took from start to finish for each run. Again, the efficiency was estimated by taking the average of 100 runs.

V. RESULTS

A. Accuracy Results

Figures 6 and 7, show the accuracy distribution for both the base *k*-means algorithm (shown in blue) and the Improved *k*-means algorithm (shown in orange) for the Iris data sets. The average, minimum, and maximum values of accuracy can be seen in Table 1 for the Iris data sets.

TABLE I
ACCURACY (%) FOR IRIS DATA SETS

Data Set	Algorithm	Average	Minimum	Maximum
Original	Base *k*	95.6	70	100
Original	Improved	91.5	70	100
Data-censored	Base *k*	93.3	70	100
Data-censored	Improved	84.6	50	100

For the original Iris data set, as shown in Figure 6, the base *k*-means algorithm has a higher average accuracy at about 95% while the Improved *k*-means algorithm has an average accuracy of about 91.5%. Both the minimum and the maximum accuracy are the same for both algorithms being 70% and 100%, respectively.

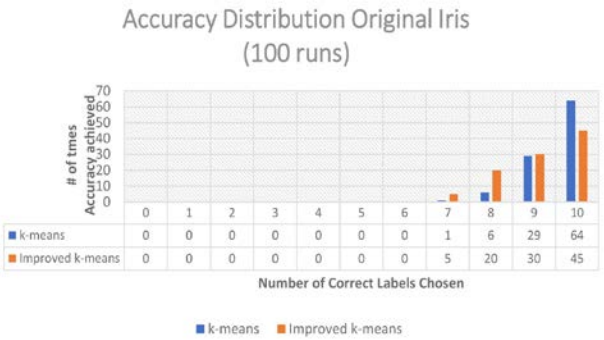

Fig 6. Accuracy distribution results from running both algorithms against the original Iris data set. The y-axis states the number of times an accuracy was achieved on a scale of 0-70 and the x-axis shows the corresponding distributions for each algorithm across a scale of 0-10.

The accuracy results for the data-censored Iris data set are shown in Figure 7. Both the base *k*-means and the Improved *k*-means algorithms showed a decreased accuracy. The base algorithm had an average accuracy of 93.3% with a minimum accuracy of 70% and a maximum accuracy of 100%. The Improved algorithm had an average accuracy of 84.6% with a minimum accuracy of 50% and a maximum accuracy of 100%.

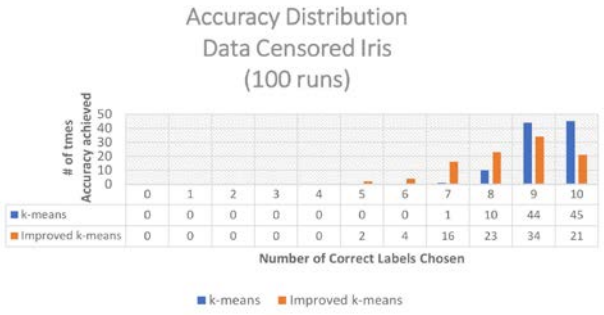

Fig 7. Accuracy distributions results from running both algorithms against the Data-censored Iris data set. The y-axis states the number of times an accuracy was achieved on a scale of 0-50 and the x-axis shows the corresponding distributions for each algorithm across a scale of 0-10.

Similarly, to the style of the Iris results, Figures 8 and 9 show the accuracy distribution for the Breast Tissue data sets and Table 2 shows the average, minimum, and maximum values of accuracy for the Breast Tissue data sets.

TABLE II
ACCURACY (%) FOR BREAST TISSUE DATA SETS

Data Set	Algorithm	Average	Minimum	Maximum
Original	Base *k*	55.5	20	100
Original	Improved	59.6	30	90
Data-censored	Base *k*	66.6	40	100
Data-censored	Improved	67.8	30	100

The accuracy results of the original Breast Tissue data set are shown in Figure 8. These results show the base algorithm had an average accuracy of 55.5% with a minimum accuracy of 20% and a maximum accuracy of 100%. The Improved algorithm had an average accuracy of 59.6% with a minimum accuracy of 30% and a maximum accuracy of 90%.

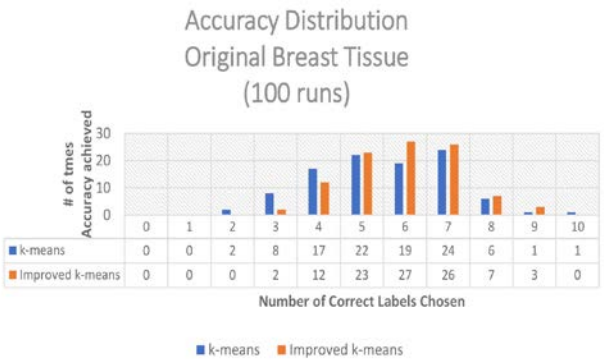

Fig 8. Accuracy distribution results from running both algorithms against the original Breast Tissue data set. The y-axis states the number of times an accuracy was achieved on a scale of 0-30 and the x-axis shows the corresponding distributions for each algorithm across a scale of 0-10.

The accuracy results of the data-censored Breast Tissue data set are shown in Figure 9. Both algorithms showed an increase in accuracy this time compared to the original data set. The base algorithm had an average accuracy of 66.6% with a minimum accuracy of 40% and a maximum accuracy of 100%. The Improved algorithm had an average accuracy of 67.8% with a minimum accuracy of 30% and a maximum accuracy of 100%.

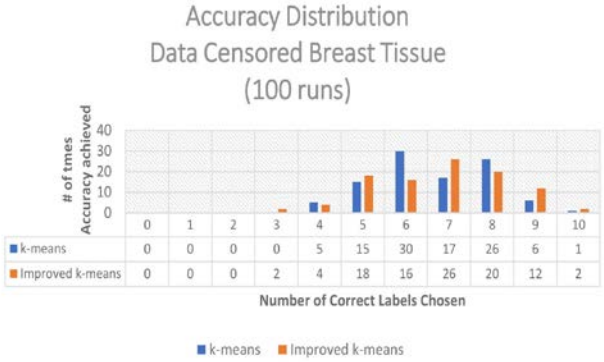

Fig 9. Accuracy distribution results from running both algorithms against the Data-censored Breast Tissue data set. The y-axis states the number of times an accuracy was achieved on a scale of 0-30 and the x-axis shows the corresponding distributions for each algorithm across a scale of 0-10.

B. Time Results

When looking at the time elapsed measurements shown in Tables 3 and 4 for each of the algorithms, it is clear to see that the Improved algorithm performs much better than the base algorithm. This is shown in the minimum, maximum, and average time calculations for each data set. The strongest contributing factor to this outcome is likely the reduced iteration count of the Improved algorithm when re-establishing centroids and closest points. This greatly compensates for the extra time spent finding the initial starting cluster centers. Best cases show the Improved *k*-means algorithm running in less than half the time of the base *k*-means algorithm. Worst cases show the Improved algorithm running in just over half the time that the base algorithm runs in.

TABLE III
TIME (SECONDS) FOR IRIS DATA SETS

Data Set	Algorithm	Average	Minimum	Maximum
Original	Base *k*	0.4162	0.2585	0.9575
Original	Improved	0.1413	0.1362	0.1677
Data-censored	Base *k*	0.4032	0.2484	0.6808
Data-censored	Improved	0.1417	0.1361	0.1798

TABLE IV
TIME (SECONDS) FOR BREAST TISSUE DATA SETS

Data Set	Algorithm	Average	Minimum	Maximum
Original	Base *k*	1.1999	0.8181	1.9074
Original	Improved	0.6480	0.6238	0.7039
Data-censored	Base *k*	1.0619	0.7926	1.6278
Data-censored	Improved	0.6377	0.6069	0.6958

VI. CONCLUSION AND FUTURE WORK

Our experiments show that the Improved *k*-means algorithm is much more efficient than the base *k*-means algorithm across all the versions of data sets that we used. In the best case, the Improved algorithm runs in less than half the time of the base algorithm, and even in the worst case, the Improved algorithm runs in just over half the time algorithm.

In terms of accuracy for both the original and the data-censored versions of the Iris data, the Improved algorithm had a lower average accuracy but could reach the same maximum accuracy that the base algorithm could. This lower average accuracy could be for a few reasons, but the main contributor is likely the fact that for the Iris data set the data points are evenly distributed among its three labels. This even distribution could produce more overlaps between data points and clusters. In the initialization method of the Improved *k*-means, centers are picked to be as far away as possible from each other. If the data points are in tight groupings or they show similarities between labels, then it can be hard to split groupings into these farthest centers accurately. With the Breast Tissue data sets, we can see that the Improved *k*-means performed more efficiently for both the original and the data-censored versions of the data set. This is probably since the Breast Tissue data set does not have an even distribution of labels.

From the results of our experimentation, we conclude that the Improved *k*-means algorithm does align with our goal, which is to improve the efficiency of the base *k*-means. However, when testing the usefulness of data-censoring, our results show this to be data set dependent. We think that finding an efficient way to choose the value of *k* at execution time should be the top priority for the progression of our algorithm. We believe that the Improved *k*-means algorithm, along with a dynamic selection of the value of *k*, could easily retain its usefulness and grow its popularity in this growing age of Big Data.

REFERENCES

[1] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, Ca, USA, 1967, pp.281-297.

[2] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering." Citeseer, 1998.

[3] P. S. Bradley, K. P. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Research, Redmond*, vol. 20, no. 0, p. 0, 2000.

[4] T. Su and J. Dy, "A deterministic method for initializing k-means clustering," in *16th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 2004, pp. 784-786.

[5] M. Yedla, S. R. Pathakota, and T. Srinviasa, "Enhancing k-means clustering algorithm with improved initial center," *International Journal of computer science and information technologies*, vol. 1, no. 2, pp. 121-125, 2010

[6] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognition*, vol. 93, pp. 95-112, 2019.

[7] D. Dua and C. Graff, "UCI machine learning repository ," 2017. [Online]. Available: http://archive.ics.uci.edu/ml