Performance Evaluation of Three Unsupervised Clustering Algorithms

Rajdeep Baruri[1], Anannya Ghosh[2]

[1]Department of Computer Science and Engineering, Jadavpur University, 188, Raja S. C. Mallick Rd, Kolkata -700 032, West Bengal, India.

[2]Institute of Engineering & Management, Y – 12, Block – EP, Sector – V, Saltlake Electronics Complex, Kolkata -700091, West Bengal, India.

ABSTRACT: Clustering is a ubiquitous technique in machine learning. Clustering is useful when we do not have labeled data. In the present study, three of the most useful and easy to implement clustering algorithms, namely k-means method, greedy-k-means method, and in the last but not the least is improved-k-means method have been studied. A study on the behavior of k-means clustering technique is being presented here. Next, we present discussion on two improved versions of k-means algorithm – in the first version, a greedy method is being applied to overcome some of the limitation whereas in the second version, using some pre-computation, we can improve the traditional k-means to some extent. While comparing the greedy version to the original to the original k-means method, our execution results suggests that the clustering quality of greedy version is better than original, but not more than that. We are not sure yet whether the size of the input dataset affects the clustering quality of the greedy version. We found that, among these three algorithms, the original k-means more-or-less performs best. While comparing the improved version to the original k-means method; the original version performs better than the improved version in most cases. For k <= 15, the improved version performs better. But as k > 15, the original version outperforms the improved version.

KEYWORDS: Machine learning, Clustering technique, Data mining, Algorithm analysis, k-Means, Cluster validity

https://doi.org/10.29294/IJASE.6.S1.2019.13-20

1. INTRODUCTION

Clustering is one kind of unsupervised learning technique in machine learning. Clustering may be useful when we do not have labelled data. K-means clustering is one of the many clustering algorithms. Clustering is the process of partitioning a set of data objects into clusters so that objects within same cluster are similar to one another yielding dissimilarities with objects within other clusters.

Clustering is widely used in many applications including business intelligence, DNA analysis in computational biology [1], security [2], geographical information system, intrusion detection [3], image retrieval, intelligent transportation system [4], music sound features analysis [5], biochemistry, social studies [6]. In this research work we are interested in only k-means clustering which is a partition-based method.

2. RELATED WORK

Though k-means is a simple clustering technique and easy to implement, there are certain factors upon which it depends heavily. Some of the common limitations are discussed below.

- Effects of outliers [7].
- k, the number of clusters [8].
- Null set of clusters [9].
- Convex shapes of clusters [10]

To improve the effectiveness and efficiency of traditional k-means clustering algorithm, a three-layer based optimization technique has been proposed [11]. In the first step, called initialization step, a top-n nearest clusters merging is performed. Then a strategy, cluster pruning, is applied to reduce the computational cost. In the third step, optimized update principle is applied.

Another brilliant effort has been applied where a greedy methodology-based constructive approach is performed [12] to reduce the clustering cost [13].

Again, another effort has been accomplished on read-world data where the analysis of k-means is done carefully. They have used two special data structures to improve execution time [14].

CLUSTERING TECHNIQUES

We have used standard Euclidean distance between two points for this purpose.

A. Lloyd's k-means clustering

Suppose we are given a dataset of n data-points where each data-point is d-dimensional. We need to cluster them into k predefined clusters so that the objective function satisfies below where S_i is any cluster and x_j is an arbitrary data-point in Si and μ_i is the center of i-th cluster:

$$\arg\min \sum_{i=1}^{k} \left(\sum_{x_j \in S_i} \| x_j - \mu_i \|^2 \right)$$

Time Complexity

The complexity of Algorithm 1 is O(nkdt) if the main loop repeats t times. Practically k << n and t << n [15].

**Corresponding Author: rajdeepbaruri2602@gmail.com*

Received: 21.05.2019 **Accepted:** 18.06.2019 **Published on**: 20.07.2019

B. Greedy k-means clustering

It is clear from the loop begins at line number 5 of Algorithm 1 that every cluster is being recomputed in each iteration. That means that we have to move a lot of data-points in each and every iteration. It may be preferable if we move just single data-point among the clusters in each iteration. What we need to find is which point makes the clustering quality best. We call that point Pbest [13]. A more conservative approach may be to move only and only Pbest.

Our execution results suggest that the clustering quality of greedy version is not better than that of the usual k-means. Sometimes this version is better than original, but not more than that. We are not sure yet whether the size of the input dataset affects the clustering quality of the greedy version. We found that, among these three algorithms, the original k-means more-or-less performs best.

C. Improved k-means clustering

While Algorithm 1 iteratively finds the partition of data-points into k clusters, it needs to calculate the distance between each and every data points in every iteration. An improvement may be to use two data structures – one to store the label of centroid within which di exists, another to store the distance to the nearest centroid from di [14], which we call DistanceOld. Algorithm 3 presents the improved version of Algorithm 1 with the help of two additional data structures: DIST and CLUSTER.

INTUITIVE IDEA

The main idea behind Algorithm 3 is that before we (re)calculate the centroids, we shall check the DistanceOld with the DistanceNew. If DistanceNew<= DistanceOld, then we do not need to calculate the pairwise distance for that di. If DistanceNew>DistanceOld, then we shall calculate the pairwise distance for that di as we were doing it same for Algorithm 1. Thus we shall be able to save execution time of original algorithm to some extent.

Algorithm 1 Lloyd K-means(D, k)

Input: D = {d_i: 1 <= i<= n} is a database containing n data-points and cj is the center of j-th cluster, 1<=j<=k.
Output: a set of k clusters
1: select k random data-objects from D as initial centroids
2: **repeat**
3: calculate the distance between all di and all cj
4: assign di to the nearest cluster
5: **for** j=1 to k **do**
6: recompute the j-th centroid
7: **end for**
8: **until** convergence criteria is met
9: **return**

Algorithm 2 Greedy K-means(D, k)

Input: D = {d_i: 1 <= i<= n} is a database containing n data-points and cj is the center of j-th cluster, 1<=j<=k.
Output: a set of k clusters
1: choose initial partition of P of k clusters randomly
2: **repeat**
3: PROFIT = 0
4: **for** j=1 to k **do**
5: find the di for which PROFIT of Cj is maximum
6: **end for**
7: **if** PROFIT > 0 **then**
8: update partition by moving di to Cj
9: **else**
10: **return**
11: **end if**
12: **until** convergence criteria is met

Method1 Profit Calculation(Cj)

1: **for** s = 1 to k **do**
2: **if**s != j **then**
3: find the cost Cost I after moving di from Cs to Cj
4: **end if**
5: find ProfitI = Current Cost – Cost I
6: **end for**
7: **return**

Algorithm 3 Improved K-means(D, k)

Input: D = {d_i: 1 <= i<= n} is a database containing n data-points and cj is the center of j-th cluster, 1<=j<=k.
Output: a set of k clusters
1: INITIALIZATION(D,k)
2: recalculate the centroids
3: **repeat**
4: **for**i=1 to n**do**
5: find the distance between di and its new centroid
6: **if**newDistance<= DIST[i] **then**
7: stay
8: **else**
9: compute newDistance among di and all centroids
10: move di to nearest cluster Cx
11: update CLUSTER[i]
12: update DIST[i]
13: **end if**
14: **end for**
15: **until** convergence criteria is met

3. EXPERIMENTAL SETUP

We have performed a series of experiments on a Lenovo ThinkPad E460 Ultrabook running the 64-bit Windows 10 Pro. The value of k has been varied from 3 to 20. Each simulation has been tested for 12 times and the average result is taken.

3.1 Datasets

We have performed our experiments on four real-world datasets available from the UCI Machine Learning Repository [16]. Table 2 represents the datasets in ascending order of their sizes. The first column represents the symbolic names.

3.2 Performance Evaluation Criteria

In order to evaluate the quality of the clustering, we introduce four basic coefficients, namely silhouette index (SI), sum of error (SSE), Davies-Bouldin index (DBI), and Dunn Index (DI). Table II shows the parameters of the four real-world datasets ordered by increasing size.

Method 2 Initialization(D,k)

1: choose k data-points randomly as initial centroids
2: **for**i = 1 to n**do**
3: calculate the distance between di and all Cj
4: set DIST[i] = minimum value in the previous step
5: assign di to the nearest cluster Cg
6: set CLUSTER[i] = g
7: **end for**
8: **return**

TABLE 1 ALGORITHMS WE HAVE EXPERIMENTED WITH

Sl No.	Name of the Algorithm	Nature of the Algorithm
A01	Algorithm 1	Iterative
A02	Algorithm 2	Greedy
A03	Algorithm 3	improved

TABLE 2 DESCRIPTION OF REAL WORLD DATASETS

Sl No.	Name of the Dataset	Size in KB	Number of Instances	Number of Attributes
D01	Iris	4.4	150	4
D02	Wine	10.5	178	13
D03	Glass	11.6	214	10
D04	Ecoli	19	336	8

TABLE 3 DESCRIPTION OF CLUSTER VALIDATION METRICS

Sl No.	Name of the Metric	Requirement for optimal
M01	Silhouette Index	Higher
M02	Davies-Bouldin Index	Lower
M03	Dunn Index	Higher
M04	Sum of Squared Error	lower

D. Silhouette Index

For a given cluster Cj, the silhouette width, si, is defined as below where ai is the average distance between the ith sample and all of the samples included in Cj and bi is the minimum average distance between the ith sample and all of the samples clustered in Cy, y = 1,2, …, k, y != j.

$$si = \frac{bi - ai}{\max\{ai, bi\}}$$

The *global silhouette index* [17], denoted by GSu, can be used as a validity index for a partition U.

$$GSu = \frac{1}{k}\sum_{j=1}^{k} sj$$

E. Davies-Bouldin Index

Let vl and vm be the centroids for l-th and m-th cluster respectively. Let xl denotes any arbitrary point within l-th cluster. Let Sl be the intra-cluster scatter. Let Dlm be the distance between two clusters whose centroids are vl and vm. Let Rlm denotes the joint cluster scatter over the distance between cluster l and m. Rlm can be calculated as:

$$Rlm = \frac{(Sl + Sm)}{Dlm}$$

The Davies-Bouldin index is defined as

$$DBI = \frac{1}{k}\sum_{l=1}^{k} Rl$$

Where

$$Rl = \max_{l \in k, l \neq m} Rlm$$

A minimal value of DBI indicates an optimal k [18].

F. Dunn Index

Let ΔS denotes the maximum distance between two data-points in Cl.

$$\Delta S = \max_{x,y \in S}\{d(x, y)\}$$

Let Ck be another cluster. The inter-cluster distance between Cl and Ck, denoted by $\delta(l, k)$, is the smallest distance between all pair of points where one point belongs to l and the other point belongs to k.

Dunn index [19] is defined as

$$\frac{\min\{\Delta S\}}{\max\{\delta(S, T)\}}$$

G. Sum of Squared Error

Let pi be an arbitrary data-point inside j-th cluster Cj. Let the centroid of Cj be denoted by mj. Now distance between point pi and cluster Cj is defined as ||pi-mj||. The sum of squared error (SSE) is defined as

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{n} ||pi - mj||^2$$

We have used python programming language version 2.7.14 with pandas 0.21.0 version and NumPy 1.51.1 version. We have compared the quality of the cluster generated using four validation metrics. We have used the *timeit* module to measure the time to execute of each of these algorithms. Due to space limitation we skip the implementation details, we focus only on the results obtained. Algorithm 1, Algorithm 2, Algorithm 3 is represented by the blue, red and green line respectively. X-axis represents the value of k, y-axis represents the mentioned validity metric. Table III shows the metrics we have used; last column shows the requirement for optimal cluster pattern.

4. RESULTS OF EVALUATION AND ANALYSIS

It is clear from the graphical outputs that Algorithm 1 performs best among the three algorithms and Algorithm 2 performs worst. Sometime the green line seems to be better than blue lines as suggested by Fig. 1. Surprisingly in Fig 2, initially the cluster quality appears best as the green line stays up enough high. But as soon as the value of k increased by our program, the quality of the cluster gets degraded. Similar conclusion can be drawn from Fig. 5 that lower k value implies better cluster quality. The size of D04 is much greater than that of D03, as suggested from Table II. We conclude that for larger dataset, Algorithm 1 preferable to us. That is the original clustering algorithm outperformed the other two.

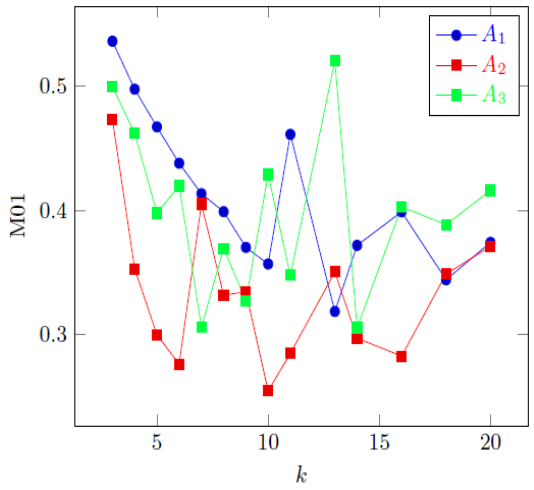

Figure 1. M01 on D01

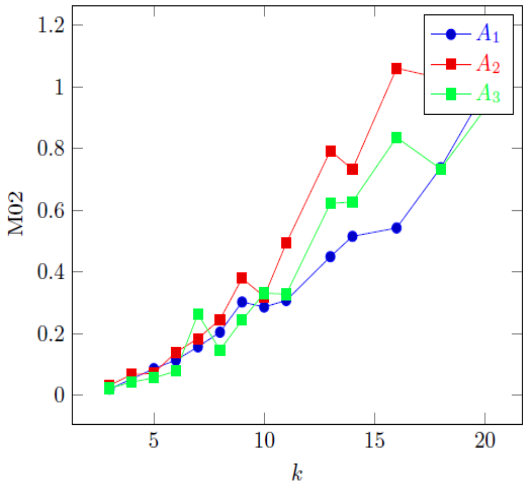

Figure 2. M02 on D01

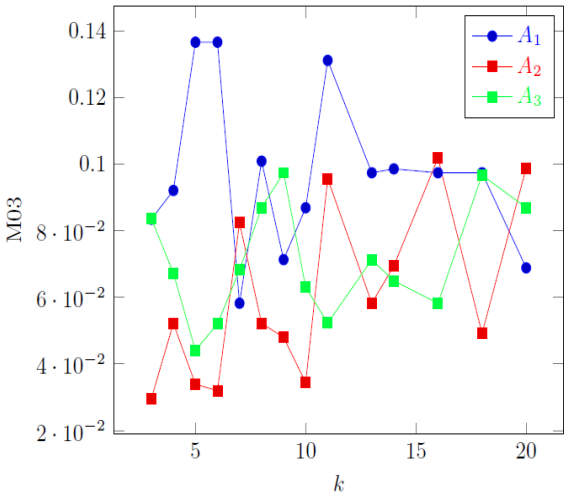

Figure 3. M03 on D01

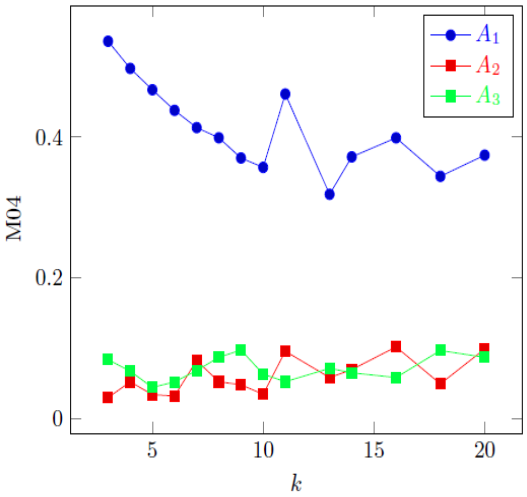

Figure 4. M04 on D01

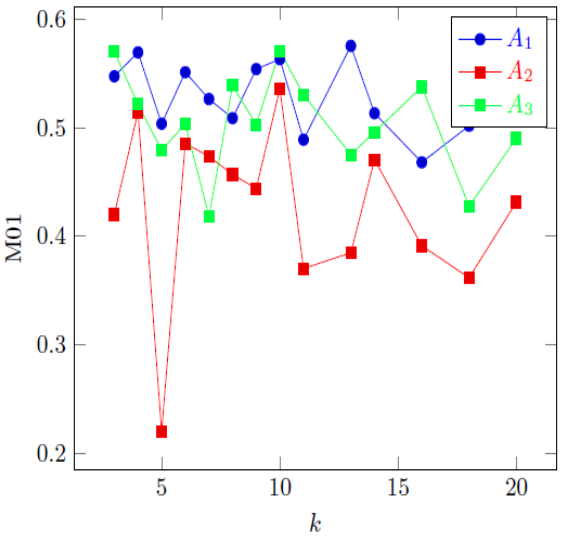

Figure 5. M01 on D02

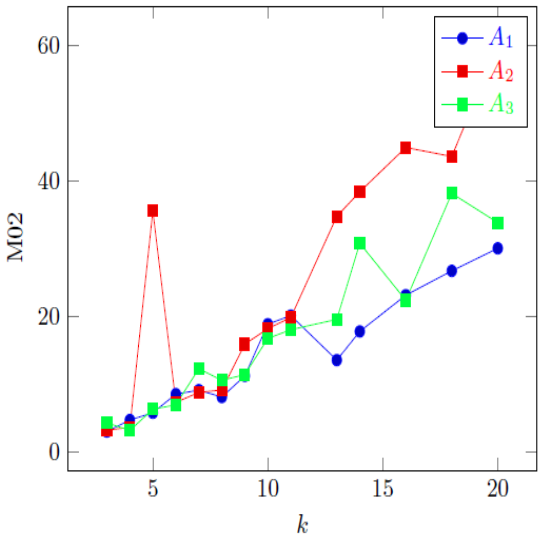

Figure 6. M02 on D02

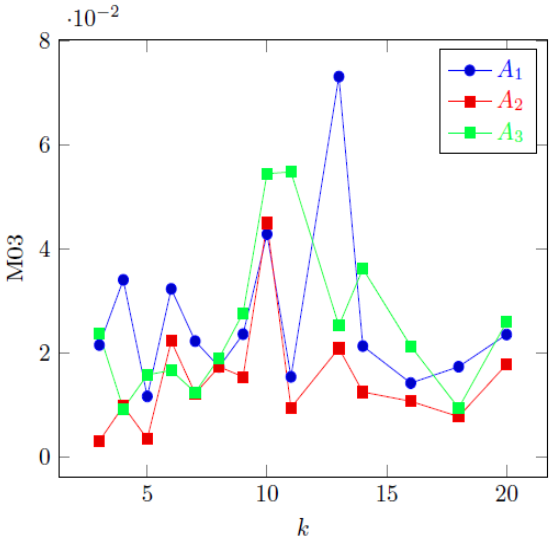

Figure 7. M03 on D02

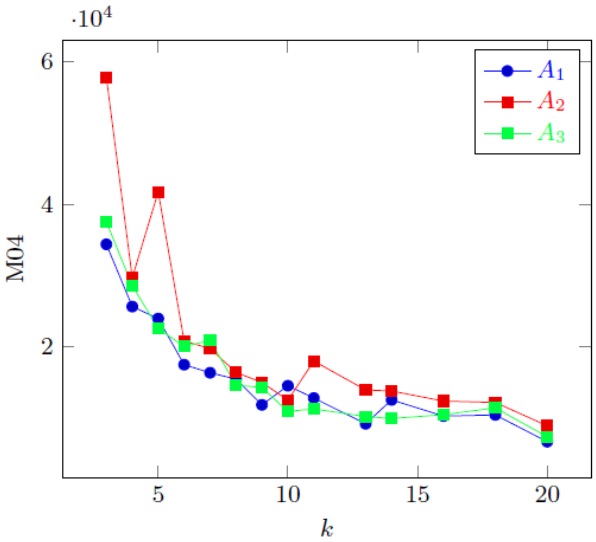

Figure 8. M04 on D02

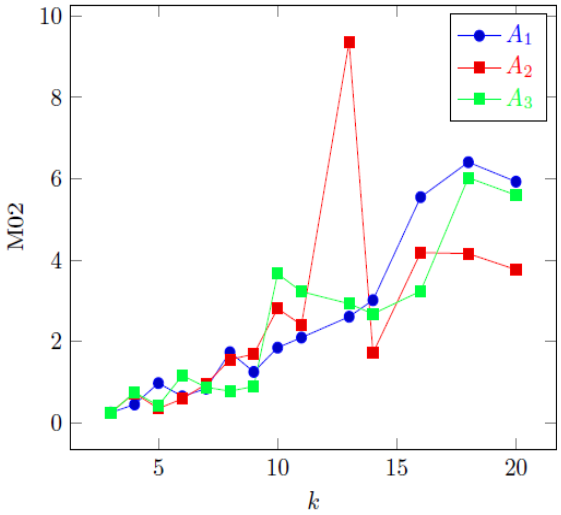

Figure 9. M02 on D03

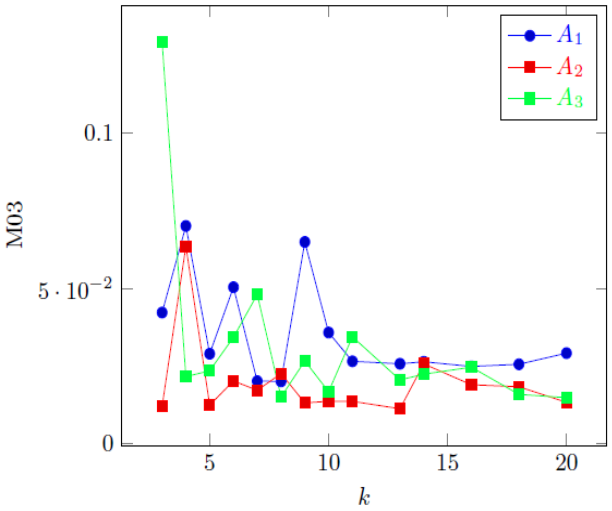

Figure 10. M03 on D03

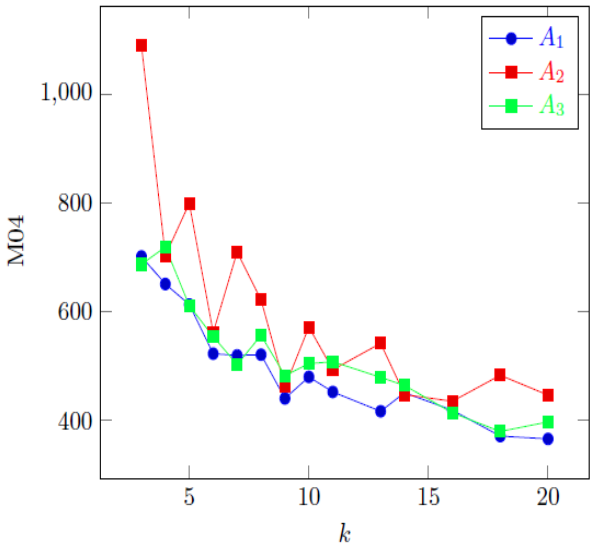

Figure 11. M04 on D03

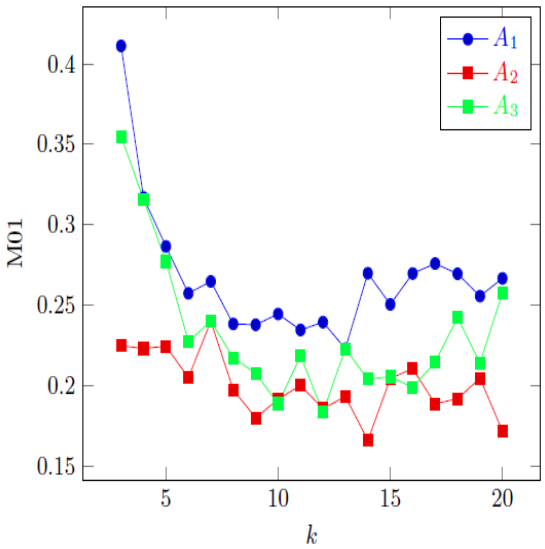

Figure 12. M01 on D04

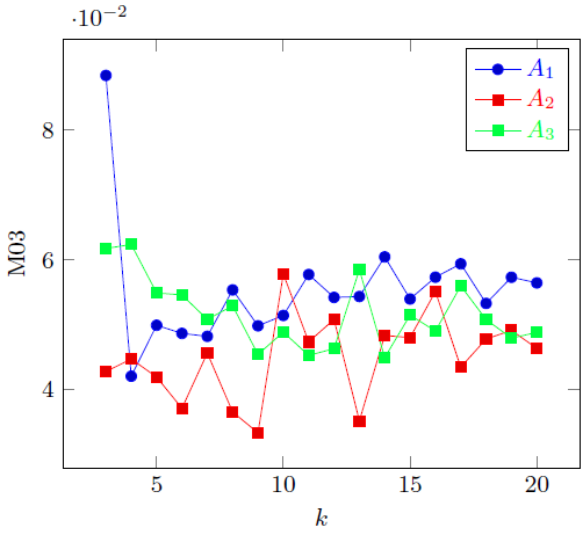

Figure 13. M03 on D04

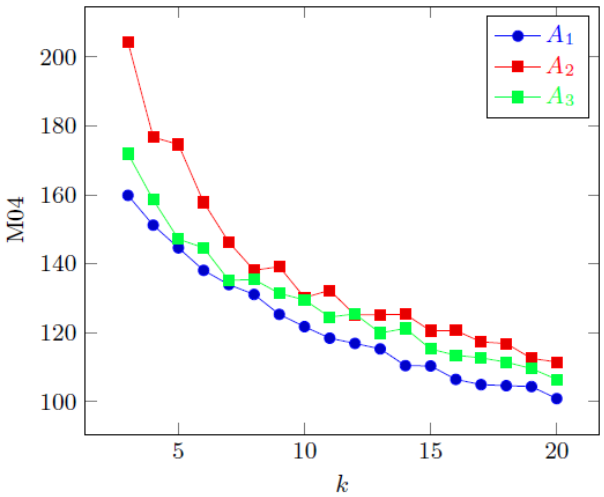

Figure 14. M04 on D04

5. FUTURE RESEARCH

We can say that sometimes Algorithm 3 performs best. We are interested to find the reason behind that. We have used array data structure as the auxiliary data structure for that algorithm. It may be interesting to find out whether the use of heap, stack, or may be queue instead of simple array can affect the execution time or not. Bayesian Information Criterion [20] is an alternative method for detecting the number of optimal number clusters. It may an interesting choice to study the behavior of Bayesian Information Criterion and compare the results with k-means clustering methods. Moreover, k-strange point algorithm [15] is another choice of traditional k-means algorithm.

6. CONCLUSION

Algorithm 1 is the best algorithm among the three, for dataset we have selected. For small k, Algorithm 3 may perform quite better, but as k> 15, Algorithm 1 starts performing faster, as suggested by Fig. 8 to Fig 14. What about the results when k > 20 or even k > 50, is yet to be seen in further research.

REFERENCES

[1]. Chormunge, S., Jena, S., 2014. Evaluation of Clustering Algorithm for High Dimensional Data Based on Distance Functions. International Conference on Information and Communication Technology for Competitive Strategies.

[2]. Li, W., 2008. Modified k-means Clustering Algorithm. Congress on Image and Signal Processing, pp. 618–621.

[3]. EslamnezhadM.,Varjani, A., 2014. Intrusion Detection Based on MinMax k-means Clustering. 7th International Symposium on Telecommunications, pp. 804-808.

[4]. Nath, R. P. D., Lee,H.-J., Chowdhury,N. K., Chang,J.-W., 2010. Modified k-means Clustering for Travel Time Prediction based of Historical Traffic Data. International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 511 – 521.

[5]. Krey, S.,Ligges, U.,Leisch, F., 2014. Music and Timbre Segmentation by Recursive Constrained k-means Clustering. Computational Statistics, pp. 37-50.

[6]. Wasserman, S., Faust, K., 1994. Social Network Analysis: Methods and Applications, ser. Structural Analysis in the Social Sciences, Cambridge University Press.

[7]. Wu, X., Kumar,V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng,A., Liu,B.,yu,P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., Steinberg, D., 2008. The Top 10 Algorithms in Data Mining. Knowledge and Information System, 1-37.

[8]. Maulik, U., Bandyopadhyay, S., 2002. Performance Evaluation of Some Clustering Algorithm and Validity Indices. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1650-1654.

[9]. Abbas, O. A. 2008. Comparisons Between Data Clustering Algorithms. International Arab Journal of Information Technology.

[10]. Patil, Y. S., Vaidya, M. B., 2012. A Technical Survey on Cluster Analysis in Data Mining. International Journal of Engineering technology and Advance Engineering, 502-513.

[11]. Qi, J., Yu, Y., Wang, L., Liu, J., 2016. K*-means: An Effective and Efficient k-means Clustering Algorithm. IEEE International Conference on Big Datas and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom),pp. 242-249.

[12]. Wilkin, G. A., Huang, X., 2007. K-means Clusteing Algorithms: Implementation and Comparison. Proceedings of the Second

International Multi-Symposiums on Computer and Computational Sciences, pp. 133-136.
[13]. Jones, N. C.,Pevzner, P. A., 2004. An Introduction to Bioinformatics Algorithms. The MIT Press.
[14]. Na, S., Xumin, L., Yong, G., 2010. Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm. Proceedings of the Third International Symposium on Intelligent Information Technology and Security Informatics, pp. 63-67.
[15]. Johnson, T., Singh, S. K., 2015. k-strange Points Clustering Algorithm. Proceedings of the International Conference on Computational Intelligence in Data Mining. Springer, pp. 415-425.
[16]. Dheeru, D.,Taniskidou, E. K., 2017. UCI machine learning repository.
[17]. Rousseevw, P. J., 1987. Silhouttes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics, pp. 53-65.
[18]. Davies, D. L., Bouldin, D. W., 1979. A Cluster Separation Measure. IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 224-227.
[19]. Dunn, J. C., 1974. Well-Separated Clusters and Optimal Fuzzy Partitions. Journal of Cybernetics, pp. 95-104.
[20]. Zaho, Q., Hautamaki, V., Franti, P., 2008. Knee Point Detection in BIC for Detecting the Number of Clusters. Advanced Concepts for Intelligent Vision Systems, pp. 664-673.

Selection and/or Peer-review under the responsibility of 2nd International Students' Conference on Innovations in Science and Technology (Spectrum – 2019), Kolkata