Research Article

A Method of Data Clustering for Detecting Outlier from K-Means Clusters

Muhammad Shaheen[1]* and Abdullah[2]

[1]Faculty of Engineering and Information Technology, Foundation University Islamabad, Pakistan; [2]National University of Computer and Emerging Sciences, Peshawar, Pakistan.

Abstract: Classification in data mining is one of the major functionalities that is performed either by predicting the value of unknown class labels on the basis of previously labeled data or to make groups of the dataset on the basis of some implicit similarity measure. Clustering works on unsupervised datasets and converts datasets to groups on the basis of some measures like Euclidean distance in K Means Clustering. The performance of K Means can significantly be affected by outliers. Outliers are not dealt in the K Means algorithm. This paper proposes a change in the K Means algorithm to accommodate the method for outlier detection on the basis of the threshold value. The threshold value of the outlier named as clus_span is computed by taking distance of each point from each other point and dividing it by the total number of points. All the points of a dataset that do not qualify the value of the minimum threshold are considered as outliers. New K Means with this add-in is tested on benchmark dataset for identification of outliers and compared with the existing K means algorithm in terms of accuracy. An improvement in performance is evident.

Received: January 11, 2021; **Accepted:** 21 November 2021; **Published:** 28 December 2021
***Correspondence:** Muhammad Shaheen, Faculty of Engineering and Information Technology, Foundation University Islamabad, Pakistan; **Email:** dr.shaheen@fui.edu.pk
Citation: Shaheen, M. and Abdullah. 2021. A method of data clustering for detecting outlier from K-means clusters. *Journal of Engineering and Applied Sciences,* 40(2): 127-134.
DOI: https://dx.doi.org/10.17582/journal.jeas/40.2.127.134
Keywords: K means, Clustering, Outlier, Algorithm, Pivot, Global outliers, Centroid, Initial seeds

Introduction

Data in its raw form is not meaningful enough to help users in making decisions for which data is processed to be stored in the form of information. In the last few decades, it was realized that if data can be converted to knowledge it will assist decision makers in reaching reliable and wise decisions. Data Mining is the domain in which data is mined to extract hidden trends to discover knowledge. The patterns which are extracted by data mining are implicit and invisible in raw data. Wise decisions depend upon such analytical insights. The process of data mining is carried out in a series of steps starting from storing data to database, transiting through data warehousing, data integration and concluding at patterns extracted after applying some algorithms. Multiple models which are not limited to CRISP-DM (Shearer, 2000), SEMMA (https://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm) are proposed for giving a framework to the process of data mining. In Figure 1 Process of knowledge discovery is modeled that includes selection of relevant data from large data repositories, then data is preprocessed and transformed into required format and hidden insights are drawn from that data and knowledge is discovered in the last phase. CRISP-DM is a standard process of Data Mining with 6 phases similar to SEMMA but focus on business understanding and data understanding and deployment which is why it is applicable broadly (Shaheen *et al*., 2011b). Whereas SEMMA from SAS institute is an alternate standard process for data mining but omits deployment. The stages of Data Mining in first place does sample collection and subset dataset take place in second exploration of patterns in data. Data is cleaned in third stage and transformed in forth stage after which the best fit model, its usefulness and reliability of the results is computed (Han, 2011).

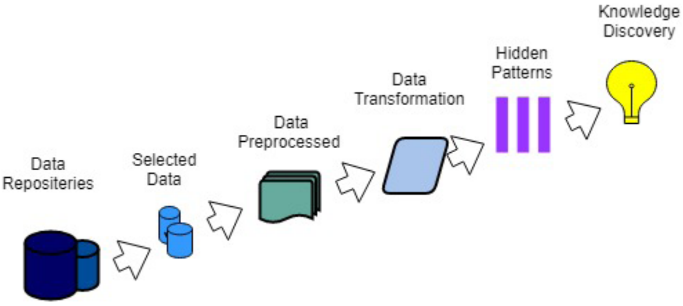


Figure 1: *Process of data mining (Han, 2011).*

The techniques through which data mining addresses the said problem (s) can broadly be classified into (1) Classification, (2) Prediction, and (3) Clustering. The classification done by data mining is done through automated or semi-automated methods which conclude in the extraction of non-trivial and previously unknown patterns (Shaheen *et al*., 2011) e.g. anomaly detection, graph to find dependencies and association rules (Hipp *et al*., 2000; Shaheen *et al*., 2013a) and multi-chain classifiers (Ali and Asghar, 2019). In clustering, the datasets are unsupervised and the techniques used for clustering are K-Means, K-Mediod, Subspace clustering, etc. (Mittal *et al*., 2019).

Placement of data points into defined groups based on previous placements is called classification. Learning greatly relates to the classification. Humans in their daily life classify animals into mammals and non-mammals, weather into cold and hot, students into intelligent and dull, etc. Classification can be done by Machine Learning as well as by Data mining (Jameel and Rehman, 2018). Classification in data mining is divided into supervised, semi-supervised and unsupervised classification (Shaheen *et al*., 2011a). When a dataset contains class labels and can be divided into training and test sets based on the presence of class labels, supervised classification is used. On the other hand, unsupervised classification is applied on the datasets without class labels (Shaheen *et al*., 2011a). Semi-supervised classification is a mixture of both and applied when data instances are partially labeled. The techniques which are commonly used in supervised classification are support vector machines, naïve Bayes classification, ID3 and C4.5 decision trees, and others (Shaheen *et al*., 2019). The techniques which are commonly used for unsupervised classification includes SOMs (Self-Organizing Maps), clustering and MM models.

Since the focus of this paper is on clustering, it is an unsupervised classification technique in which data instances are grouped on the basis of some implicit similarity measure like distance (Khan and Hogg, 2014). Data clustering is a broadly studied area in data mining and the algorithms of data clustering are broadly classified into hierarchical methods, grid-based methods and partitioning methods (Mittal *et al*., 2019). K-Means is considered to be the baseline for the rest of the clustering algorithms. K-Means, K-Medians and K-Mediod algorithms of clustering are placed in the category of partitioning algorithms. The similarity in cluster data points is computed by using different measures. The points in a cluster should be similar enough to each other (inter-cluster similarity) and maybe at maximum dissimilarity with the points of other clusters (intra-cluster dissimilarity) (Han, 2011).

K-means algorithm is a technique used for clustering. It is a simple algorithm and gives better accuracy with lower complexity. The euclidean distance of the points is taken from randomly picked cluster centers to group the points with cluster centers based on distance. The downside is that the K Mmeans algorithm won't deal with unusual data. The data that won't fall in any of the existing clusters are considered to be Outlier or Extreme value. If the dataset is noisy, outliers would affect the overall accuracy of the results. These outliers may represent noise within the dataset but they do not always. An outlier can also represent an anomaly or unexpectedness (Rehman and Belhaouari, 2021).

K-means produces better results on smaller data sets. The ultimate target of the algorithm is to find a locally optimal solution. This is done by adjusting data points and minimizing the distance from its closest cluster center (Wu *et al*., 2021). The algorithm has lower complexity and gives fairly accurate results but the accuracy of the K-Means algorithm is compromised because it doesn't take care of the outliers. In a dataset, the Outlier is a point reflects abnormal behavior and won't fit in any of the class. Noise is one of the examples of an outlier. Reasonable results can be generated if the outlier is eliminated from k means. An outlier is similar to odd man out.

In the extant literature, there does not exist a method to find outliers from K Means clusters. Some methods to detect outliers from the datasets are there which include depth-based outlier detection methods, distance-based and density-based (Huang *et al*., 2017). However, the facts revealed from the existing literature are given in the following paragraph.

In the literature, two types of outlier detection methods are given; (1) Local and (2) Global outlier detection methods (Jabbar, 2021; Kriegel *et al*., 2010). The perspective of the Global method is coarse-grained. Outliers are found at a broader level. The data that is considered as reference points in these methods are considered as a whole and clusters are formed in a single pass. Conversely, in the local outlier detection methods, smaller subsets of the original dataset are marked as reference points and the final clusters are formed at multiple passes. The approach reduces the size of the cluster and detects outliers more effectively. There are certain approaches in which the reference points vary. A brief pictorial view of both the approaches is given in Figure 2.

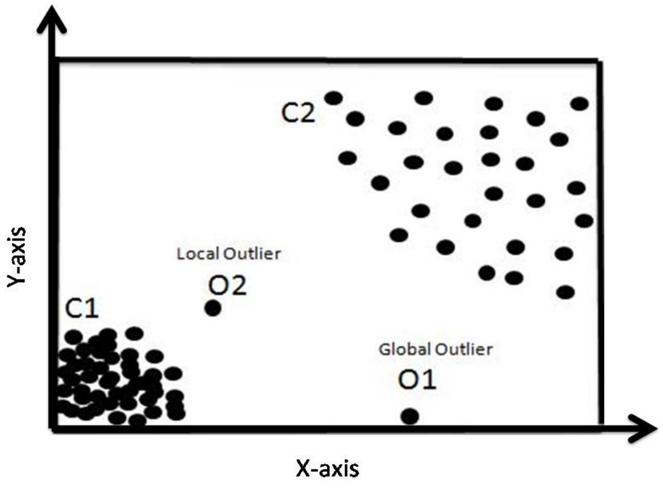


Figure 2: *Local and global outlier detection methods.*

K-Nearest Neighbor approach is effective on those clusters which are less dense. By classification of outlier detection techniques into local and global techniques, it is assumed that scope may be the only parameter for classifying outlier detection methods but this is not the case. The techniques have also been divided based on density. For outlier detection density-based methods are used, in which different density probabilities estimation strategies are used to detect outliers. Lower the density of the data point, more probable it is to be outlier (Lin *et al*., 2019). Anomaly score technique is an example of a density-based technique. Anomaly score of a data point is calculated by taking distance from a k-nearest neighbor (Wu *et al*., 2021). The anomaly score is obtained by selecting n number of nearest neighbors placed at a distance of *d* from each other. It is similar to the approach that measures the density of a data record within a hypersphere with radius r. The advantage of this method is that if the number of irregularities in the dataset are beyond some threshold, they can expose its presence in an existing cluster. Such variable density issues thoroughly remained as an area of work for researchers and may be considered as a reason for lesser diffusion of global outlier techniques. Local Outlier Factor (*LOF*) (Kotu and Deshpande, 2015) on the other hand, makes a comparison between the density of principal data point and its neighbors. The ratio of the density of points with the density of K neighbors is computed.

Local Distance-based Outlier Factor (*LDOF*) (Zhang *et al*., 2009; Wang *et al*., 2021) is based upon the calculation of distance d of a point from its neighbors in order to find scatterings or deviation of a point from its neighbors. Outliers are determined on the basis of the high score of the *LDOF*. The *LDOF* can be computed by using Equation 1.

$$LDOF_{lb} = \frac{d_{x_p}}{D_{x_p}} \quad \ldots(1)$$

In Equation 1 *d* is used for the distance between a point and its neighbors.

Outlier removal clustering (*ORC*) is another technique used for the detection and removal of an outlier (Liu *et al*., 2021). *ORC* is a two stage iterative process. In the first stage, K-means clustering will be used to classify the data points till the convergence and in the second stage, outlyingness factor of each vector will be computed. Outlyingness factor is calculated on the basis of distance from the cluster center. Like the probability factor, scaling is done on a scale of 0-1. The value which won't lie in the range of outlyingness factor is considered as outlier.

Another preliminary clustering algorithm that detects outlier cluster on the basis of Mutual Neighbors Graph (MUNG) constructed by connecting each point to its mutual neighbors and detects clusters with the idea that outlier clusters are smaller than the rest of the clusters (Huang *et al*., 2017). A survey on different outlier detection techniques whose broader outline is covered here is given in (Thudumu *et al*., 2020).

From the literature, we concluded that it is necessary to detect and remove outliers from the dataset and the techniques of outlier detection must become part of clustering techniques. If by all means the outlier is needed as part of a dataset, then its impact should be minimized to save dataset from bias value. In this paper, an outlier detection technique is proposed and added to the existing algorithm of K means clustering which modified the existing K means algorithm. If the outlier cannot be removed from the dataset, the proposed algorithm will ensure the minimization of the effect of the outliers. K means clustering algorithm is explained in the following section before going into the details of the proposed algorithm.

K-means clustering

Euclidean distance is a measure for similarity used in K Means clustering. Points with multiple coordinate are used for the representation of each attribute on the Cartesian space. A single point on Cartesian space represents the single database instance. Single variable value is represented by each coordinate of the system. Points are allocated to a k number of clusters by calculating Euclidean distance from its cluster center. At the start election of the cluster, the center is random in the K Means process. Given below are the steps for the K-Means algorithm (Shaheen *et al*., 2011a; 2013b).

(i) Pick *K* random points from a given dataset and designate them as cluster centers (CC).

$$CC..R_n$$
$$n = 1,2,,..P$$
$$CC \in [A_1, A_2, A_3, ..., A_n] \cup S_x$$

A_1, A_2, A_3, An= Data points. S_x represents new point.

(ii) Calculate distance (Euclidean distance) of every point from the cluster centers randomly picked in the previous step (Equation 2).

$$\mathrm{K_i} = \sum_{i=1}^{k}\sum_{j=1}^{n}(A_k - A_j)^2 \quad ...(2)$$

(iii) Calculate the new values of cluster centers by taking the average value of the points allocated to each cluster center (Equation 3).

$$CC_{new} = \left(\frac{1}{r}\right) \sum_{index(r)=j=1}^{n} X_{a_j}\; if\; r > 0 ...(3)$$

Here represents a point in Cartesian coordinates has all the values of associated attributes of the database.

(iv) Calculate the error by using the square error criterion (Equation 4).

$$E = \sum_{i=1}^{c} \sum_{index(p)=i=1}^{n} ||X_{a_i} - CC_{new}||\ ^2 \quad ...(4)$$

The process will run till convergence and consistency of cluster centers.

The convergence of K-Means clusters to local minima enables it to divide parse datasets into clusters based on Euclidean distance (Markov and Laroze, 2007; Shaheen *et al*., 2011a). All the data points which belong to the same cluster have statistical similarity which cannot be identified through conventional clustering techniques. The efficiency of an algorithm is determined by its time complexity which in the case of K-Means is *O (TKN)*.

N= number of input data points; *K*= number of clusters; *T*= number of iterations or passes in which K-Means converged.

The paper is organized in a systematic way. Section 2 explains the method proposed for detecting the outliers from clusters obtained by K Means clustering. In the extension of section 2, section 3 gives a modified K Means clustering algorithm. Results are presented in section 4 and section 5 concludes the paper.

Materials and Methods

In the K-means algorithm, every data point is assigned to one cluster without taking care of the fact that it might be an outlier. We proposed an algorithm to find outliers from the dataset without compromising accuracy achieved by the K-means algorithm.

For this, a variable with the name of *Clus-Span* is introduced. *Clus-Span* has the value of threshold of spanning distance of a cluster. "Outlier Threshold" is the name for this value. As this Outlier Threshold value decides how far a cluster can stretch to get an item, it is very important to calculate this value accurately. The Equation 5 is to compute OT value.

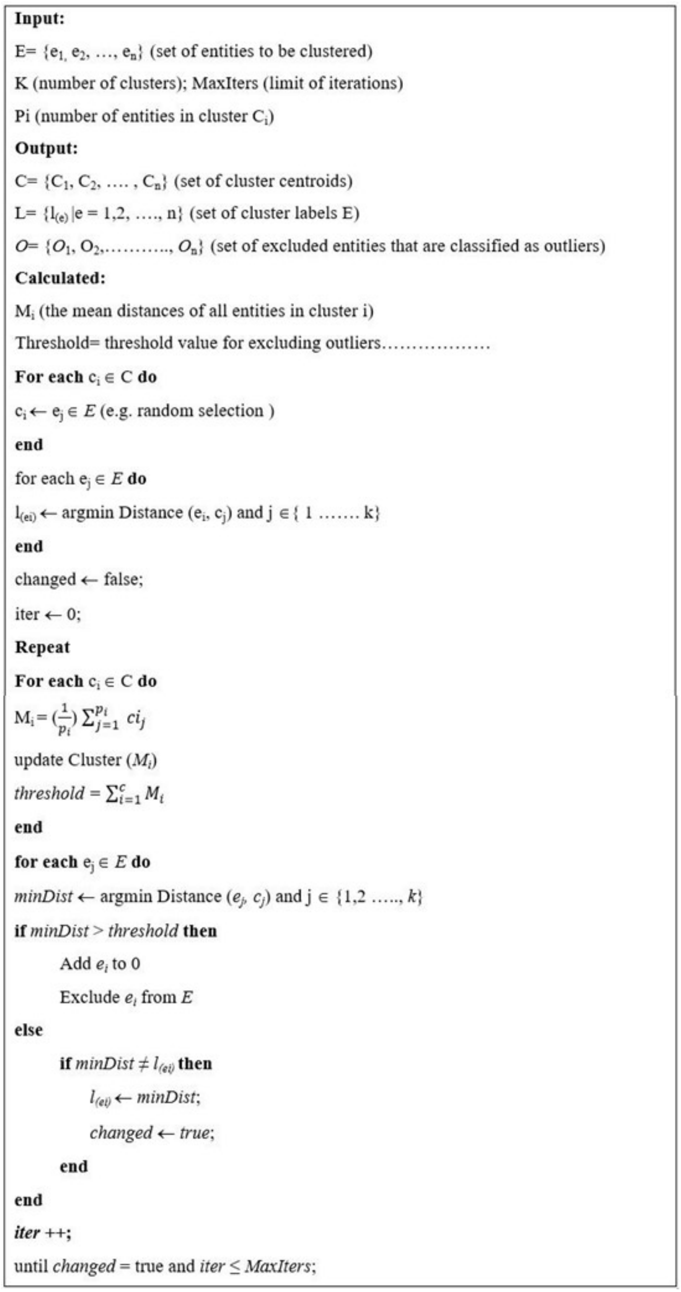
$$Clus-Span = \sum_{j=0}^{n} ⬚ \sum_{i=0}^{m} \frac{\left(\frac{\sqrt{(x_j-x_i)^2+(y_j-y_i)^2}}{m}\right)}{n} \quad ....(5)$$

Where; n= number of clusters and m = number of items in a dataset. This Equation 5 calculates the mean Euclidean distance of the points from its centroid and takes an average of mean Euclidean distance to set it as *clus-span*.

Algorithm
Name: (Outlier Detection)
Description: This algorithm is used to detect outliers from the clusters made through K means clustering algorithm.

1. Find Euclidean distance *D* as in K Mean clustering and then find the value of *clus_span* according to eq. 5.
2. The data point where *D* is greater than the value of *clus_span* will be assigned to a cluster with minimum distance
3. The data points where *D* is greater than *clus_span* will be assigned to a temporary cluster.
4. Adjust the centroid of the points allocated in Step-2 to the mean position of its points as in K mean clustering.
5. Find the value of *clus_span* again.
6. Repeat till convergence.
7. The clusters holding only one item will be deleted. A separate cluster with an outlier exists now.
8. Re-compute the distance of each item from centroids.
9. Assign the item to the closet centroid.
10. Adjust the centroid to the mean position of its points.
11. Repeat until no further assignment is possible.

The distance between data points and all the randomly picked centroids is calculated by using Euclidean distance. In step 1, euclidean distance is calculated and is compared with a threshold named *clus-span*. In step 2, those points which are below the minimum threshold are grouped in a new cluster. For the readjustment of the centroids to their original positions according to the other points of clusters, the mean of the centroid is taken in step 3. Step 4 dynamically changes the threshold value for the outlier because a new threshold value is again calculated. Till the formation of smart clusters, all steps are repeated. Once smart clusters are generated, step 6 will prune the falsely assigned data points. Finally, every data point is once again measured from the centroid to re-arrange the centroids to its actual position. The pseudo code for the algorithm is given below.



```
Input:
E= {$e_1, e_2, …, e_n$} (set of entities to be clustered)
K (number of clusters); MaxIters (limit of iterations)
Pi (number of entities in cluster $C_i$)
Output:
C= {$C_1, C_2, …. , C_n$} (set of cluster centroids)
L= {$l_{(e)}$ |e = 1,2, …., n} (set of cluster labels E)
O= {$O_1, O_2,……….., O_n$} (set of excluded entities that are classified as outliers)
Calculated:
$M_i$ (the mean distances of all entities in cluster i)
Threshold= threshold value for excluding outliers………………
For each $c_i \in C$ do
$c_i \leftarrow e_j \in E$ (e.g. random selection )
end
for each $e_j \in E$ do
$l_{(ei)} \leftarrow$ argmin Distance ($e_i, c_j$) and j ∈ { 1 ……. k}
end
changed ← false;
iter ← 0;
Repeat
For each $c_i \in C$ do
$M_i = (\frac{1}{p_i}) \sum_{j=1}^{p_i} ci_j$
update Cluster ($M_i$)
$threshold = \sum_{i=1}^{c} M_i$
end
for each $e_j \in E$ do
$minDist \leftarrow$ argmin Distance ($e_j, c_j$) and j ∈ {1,2 ….., $k$}
if $minDist > threshold$ then
        Add $e_i$ to 0
        Exclude $e_i$ from $E$
else
        if $minDist \neq l_{(ei)}$ then
            $l_{(ei)} \leftarrow minDist$;
            $changed \leftarrow true$;
        end
end
iter ++;
until $changed$ = true and $iter \leq MaxIters$;
```

(Caption removed from this place)

The dataset for testing of the algorithm is collected from two different sources. One dataset was census data and the other one was a hypothetical benchmark in which worst possible instances for the algorithm are manually added. Results of the experiments are discussed in the next section.

Results and Discussion

As mentioned earlier, the proposed algorithm is tested on two different data sets and its efficiency is compared with the efficiency of existing algorithms. One of the datasets is collected from an online resource and was census data. The other data set is a hypothetical benchmark in which a few specific data instances were added that can be considered as the worst case for the proposed algorithm.

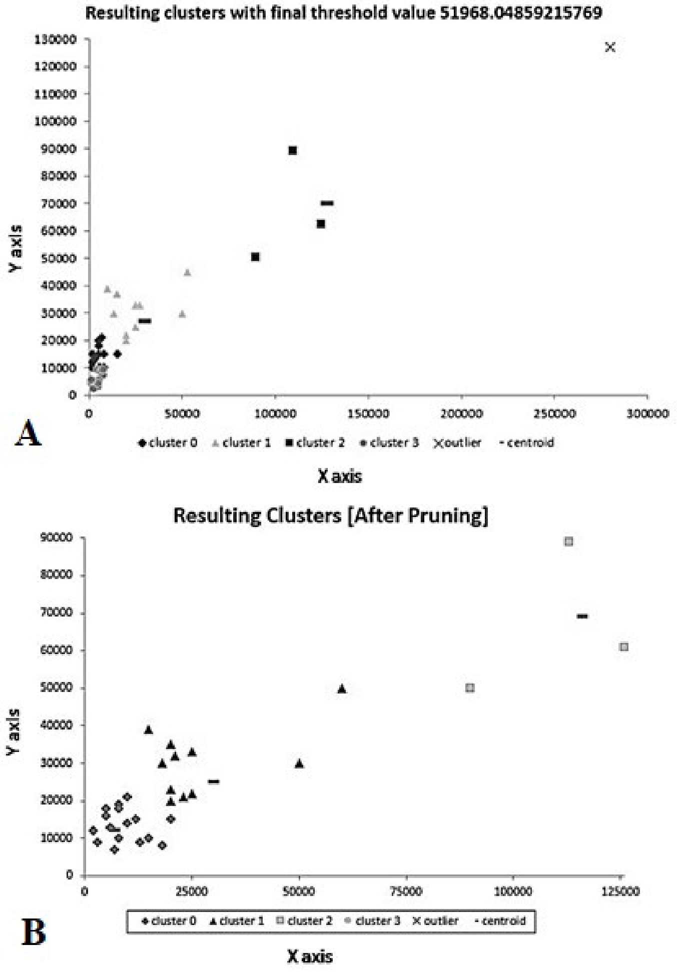


Figure 3: A: *Division of data into clusters in dataset 1;* **B:** *Resulting Clusters of dataset 1 after pruning.*

The experiment is performed by developing an add-in for WEKA software in which the add-in is plugged with an existing K Means clustering module. The experiment is performed on Intel Core 2 Duo machine placed on a distributed network. Five passes on both the datasets were run by making a slight modification in the few of the data files.

The outliers in all the passes were efficiently detected and pruned, and clusters are always being formed in noise-free data. 7% of the genuine data points were falsely classified as outliers.

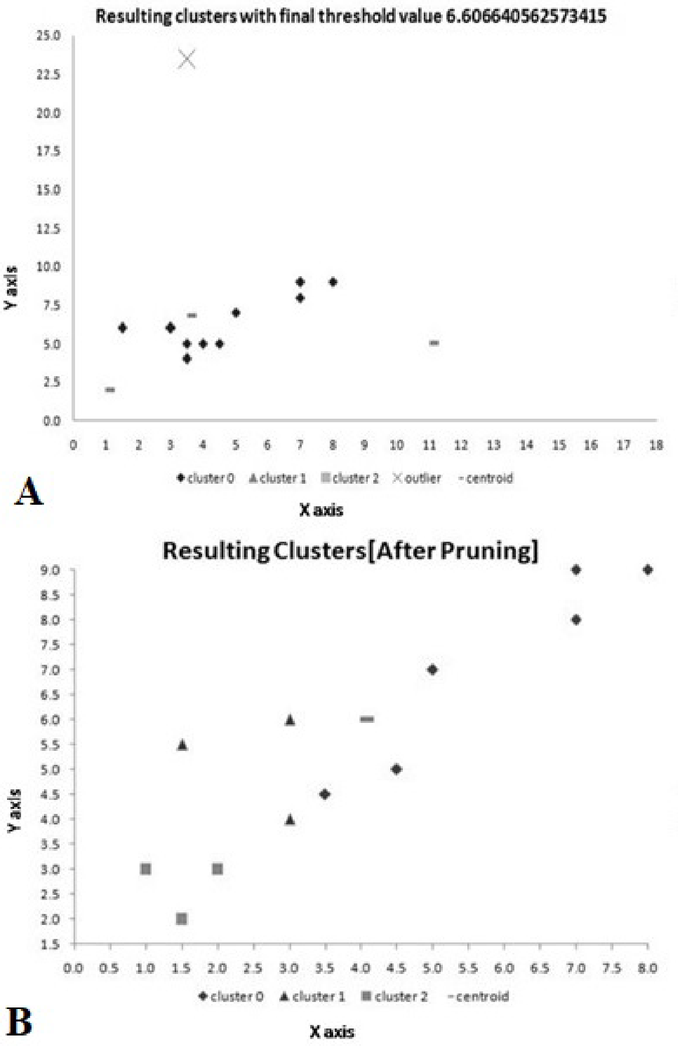


Figure 4: A: *Division of data into clusters in dataset 2;* **B:** *Resulting Clusters of dataset 2 after pruning.*

The complexity of K Means is *O (NKT)*, where *N* is the number of clusters, *K* is the number of items and *T* is the iterations till the clustering reaches equilibrium. As our algorithm does not require any extra calculations, the complexity for the first pass is still *O (ken)* while for the second pass *O (k (n-t) v)* where *t* is the number of outliers. So complexity for both passes would be *O (kv (n+ (n-t)))*. But since *t* would be a very small number it is almost negligible so we can simplify the complexity to *O (2knv)*. Experimental results on different datasets are given below in Figures 3-5. Figure 3a illustrates thresholding clusters of dataset 1 are formed and

Outlier cluster of dataset 1 is pruned as shown in Figure 3b. Same results are shown in Figures 4a, b, 5a, b on dataset 2 and dataset 3.

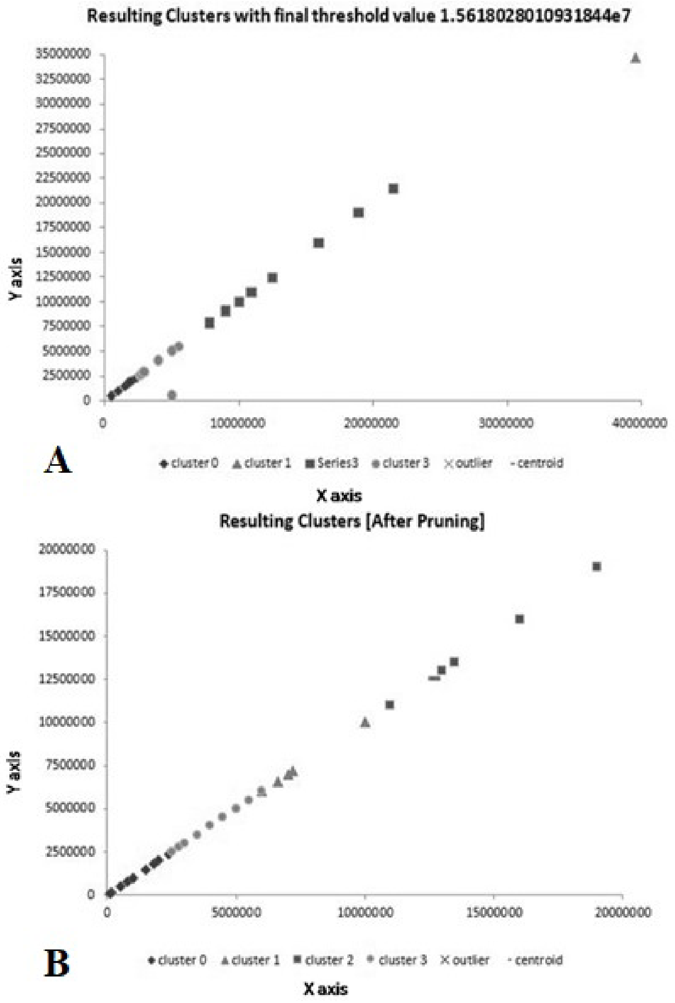


Figure 5: A: *Division of data into clusters in dataset 3;* **B:** *Resulting Clusters of dataset 3 after pruning.*

In K-Mediod, the outliers are not dealt with, but it reduces the effects of extreme values in resulting clusters by using median as centroids. This prevents outliers to be chosen as centroids resulting in inefficient clustering. But outliers are still clustered nonetheless. Our algorithm effectively removes outliers from being clustered and also is not affected by extreme values since they are pruned.

Furthermore, in the ODIN method and OCI method, the threshold value is initially assigned which could probably take many iterations of the algorithm to judge the proper value for threshold in order to get more accurate results.

Conclusions and Recommendations

K-Means clustering is the most widely used clustering technique with a drawback that it does not have the provision of outlier detection and removal. The paper proposed an outlier detection algorithm which after experimentation is found to be efficient in terms of accuracy and execution time and yields better results in removing and detecting outliers. The time complexity of the algorithm is linear as compared to other outlier detection techniques. All asymptotic complexities for the algorithms are same.

The work can be extended to include different parameters like size of the cluster, concentration of cluster points (called as density) and spread of the cluster, in calculation of the value of outlier threshold *clus-span*. Selection of initial centroids on the basis of certain criterion can also improve the efficiency of the algorithm.

Novelty Statement

This manuscript warrants original and novel contribution to the domain of clustering in data mining.

Author's Contribution

Muhammad Shaheen: Conducted the research and wrote the article.
Abdullah: Helped in researcha nd write up.

Conflict of interest

The authors have declared no conflict of interest.

References

Ali, T., and S. Asghar. 2019. Efficient label ordering for improving multi-label classifier chain accuracy. J. Natl. Sci. Found. Sri Lanka, 47(2): 175-184. https://doi.org/10.4038/jnsfsr.v47i2.9159

Chandola, V., A. Banerjee and V. Kumar. 2009. Anomaly detection: A survey. ACM Comput. Surv., 41(3): Article 15. https://doi.org/10.1145/1541880.1541882

Han, J., 2011. Data mining: Concepts and techniques. 3rd Edition: Morgan Kaufman Publishers. pp. 12-40.

Hipp, J., U. Güntzer and G. Nakhaeizadeh. 2000. Algorithms for association rule mining: A general survey and comparison. ACM Sigkdd Explor. Newsl., 2(1): 58-64. https://doi.org/10.1145/360402.360421

Huang, J., Q. Zhu, L. Yang, D.D. Cheng and Q. Wu.

2017. A novel outlier cluster detection algorithm without top-n parameter. Knowl. Based Syst., 121: 32-40. https://doi.org/10.1016/j.knosys.2017.01.013

Jabbar, A.M., 2021. Local and global outlier detection algorithms in unsupervised approach: A review. Iraqi J. Elect. Electron. Eng., 17(1): 10.37917/ijeee.17.1.9. https://doi.org/10.37917/ijeee.17.1.9

Jamil, S., and S. Rehman. 2018. An optimal feature selection method using a modified wrapper based ant colony optimization. J. Natl. Sci. Found. Sri Lanka, 46(2): 143-151. https://doi.org/10.4038/jnsfsr.v46i2.8414

Khan, N.A., and D. Hogg. 2014. Unsupervised learning of appearance classes from video. NED Univ. J. Res. 11(2): 15-26.

Kotu, V., and B. Deshpande. 2015. Anomaly detection. Predictive analytics and data mining concepts and practice with rapidminer. pp. 329–345. https://doi.org/10.1016/B978-0-12-801460-8.00011-2

Kriegel, H.P., P. Kröger, E. Schubert and A. Zimek. 2010. Interpreting and unifying outliers scores. In: Proceeding of 11th SIAM International Conference on Data Mining. Mesa, AZ. pp. 13-24. https://doi.org/10.1137/1.9781611972818.2

Lin, C.H., K.C. Hsu, K.R. Johnson, M. Luby and Y.C. Fann. 2019. Applying density-based outlier identifications using multiple datasets for validation of stroke clinical outcomes. Int. J. Med. Inf., 132(103988): 10.1016/j.ijmedinf.2019.103988. https://doi.org/10.1016/j.ijmedinf.2019.103988

Liu, H., J. Li, Y. Wu and Y. Fu. 2021. Clustering with outlier removal. IEEE Trans. Knowl. Data Eng., 33(6): 2369–2379. https://doi.org/10.1109/TKDE.2019.2954317

Markov, Z., and D.T. Larose. 2007. Data mining the web: Uncovering patterns in web content, structure and usage. Wiley Publishers USA. pp. 192-194. https://doi.org/10.1002/0470108096

Mittal, M., L.M. Goyal, D.J. Hemanth and J.K. Sethi. 2019. Clustering approaches for high dimensional databases: A review. Wiley data mining and knowledge discovery. https://doi.org/10.1002/widm.1300

Rehman, A., and S.B. Belhaouari. 2021. Unsupervised outlier detection in multidimensional data. J. Big Data, 8(80): https://doi.org/10.1186/s40537-021-00469-z

Shaheen, M., S. Iqbal and F. Basit. 2013b. Labeled clustering: A method to label unsupervised classes. In: Proceedings of 8th International Conference on Internet Technology and Secured Transactions, United Kingdom: 210-214. https://doi.org/10.1109/ICITST.2013.6750193

Shaheen, M., M. Shahbaz, A. Guergachi and Z. Rehman. 2011b. Data mining applications in hydrocarbon exploration. Artif. Intell. Rev., 35(1): 1-18. https://doi.org/10.1007/s10462-010-9180-z

Shaheen, M., M. Shahbaz, A. Guergachi and Z. Rehman. 2011a. Mining sustainability indicators to classify hydrocarbon development. *Knowl. based Syst.,* 24: 1159-1168. https://doi.org/10.1016/j.knosys.2011.04.016

Shaheen, M., M. Shahbaz and A. Guergachi. 2013a. Context based positive and negative spatio temporal association rule mining. Knowl. Based Syst., 37: 261-273. https://doi.org/10.1016/j.knosys.2012.08.010

Shaheen, M., T. Zafar and A.S. Khan. 2019. Decision tree classification: Ranking journals using IGIDI. J. Inf. Sci., 46(3): https://doi.org/10.1177/0165551519837176

Shearer, C., 2000. The CRISP-DM model: The new blueprint for data mining. J. Data Warehouse., 5: 13-22.

Thudumu, S., P. Branch, J. Jin and J. Singh. 2020. A comprehensive survey of anomaly detection techniques for high dimensional big data. J. Big Data, 7(42): https://doi.org/10.1186/s40537-020-00320-x

Wang, R., Q. Zhu, J. Luo and F. Zhu. 2021. Local dynamic neighborhood based outlier detection approach and its framework for large-scale datasets. Egypt. Inf. J., 22(2): 125–132. https://doi.org/10.1016/j.eij.2020.06.001

Wu, C., B. Yan, R. Yu, B. Yu, X. Zhou, Y. Yu and N. Chen. 2021. K-Means Clustering Algorithm and Its Simulation Based on Distributed Computing Platform. Complexity, 9446653: https://doi.org/10.1155/2021/9446653

Zhang, K., M. Hutter and H. Jin. 2009. A new local distance-based outlier detection approach for scattered real-world data. In: Proceeding of 13th Pacific-Asia conference on knowledge discovery and data mining. pp. 813-822. https://doi.org/10.1007/978-3-642-01307-2_84