Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

Modified K-Means Clustering with Semi Grouping Perspective : A Study

Said Al Afghani Edsa[1], Gerry Chandra[2]
said.afghani@kbij.co.id, gerry.chandra@kbij.co.id
[1]Credit Bureau Indonesia
[2]Advanced AI

Article Information

Submitted: 24 Jan 2023
Reviewed: 16 Mar 2023
Accepted : 20 Apr 2023

Keywords

K-Means, Accuracy, Semi Grouping, Clustering

Abstract

We will provide some results of a literature study related to one of the clustering methods, namely K-Means, but with some modifications, devoted to the case of computation time. Modifications were made at the time of determining the cluster center by previously applying principal component analysis (PCA), other researchers [4] proposed this method first, which differs in this note, namely in the preprocessing of the data before principal component analysis is carried out. Comparison of the accuracy of the cluster results is also given in this note.

A. Introduction

In principle, clustering aims to group data into several clusters, of course more than one cluster, with the characteristic that intra-clusters have high similarities and inter-clusters have low similarities. In some cases, clustering can help us in order to customer segmentation, grouping of employee performance, or to check the existing of anomalies in our dataset . There are several clustering algorithms that have been developed both in terms of heuristics and hierarchies. One of them is the K-means (heuristic) using the euclidean distance principle [2]. In a practical scope, the need for fast computational processes is very necessary, for this reason in this article we study a way to cut this time process related to K-Means clustering, namely when determining the center of clustering, where in the tradition k-means of determining the centroid at first taken randomly, and the iterative process of determining centroids and clusters continues until they converge. In this study, the determination of the initial centroid is not done randomly but rather through grouping with firstly ordering the dataset then using principal component analysis and finally using percentiles to determine the centroid, so that the dataset is actually "grouped" which means that when the next grouping process is carried out it can cut down the processing time -using the K-Means algorithm.

B. Research Method

Several studies related to clustering with several approaches have been carried out, along with some conclusions that have been obtained by each researcher in relation to the dataset used [1]. Especially in K-Means clustering, it is known that the determination of the cluster center is taken randomly, using the distance principle, namely the Euclidean distance, calculating the distance between the initial cluster center and the objects around it, so that the object with the closest distance to the initial cluster center will be at the same group, the next process is the determination of a new cluster center, obtained by calculating the average of the initial cluster formed, the process of calculating the distance and updating the cluster center again until the cluster center does not change again or in other words it has reached convergence. In this article, the author tries to offer a principle in which the process of determining the initial cluster is not done randomly. The idea is to first arrange the data points so that adjacent points will be easy to group later, then the initial cluster determination is taken from the average data that has been grouped, for example using a percentile. In practice, first the data is sorted, then the main components are taken and grouped using percentiles, then the average point of the data that has been grouped is searched, namely as the center of the cluster,

and finally clustering is done using K-Means on the data that we have processed. The process flow of our method is like the following picture:

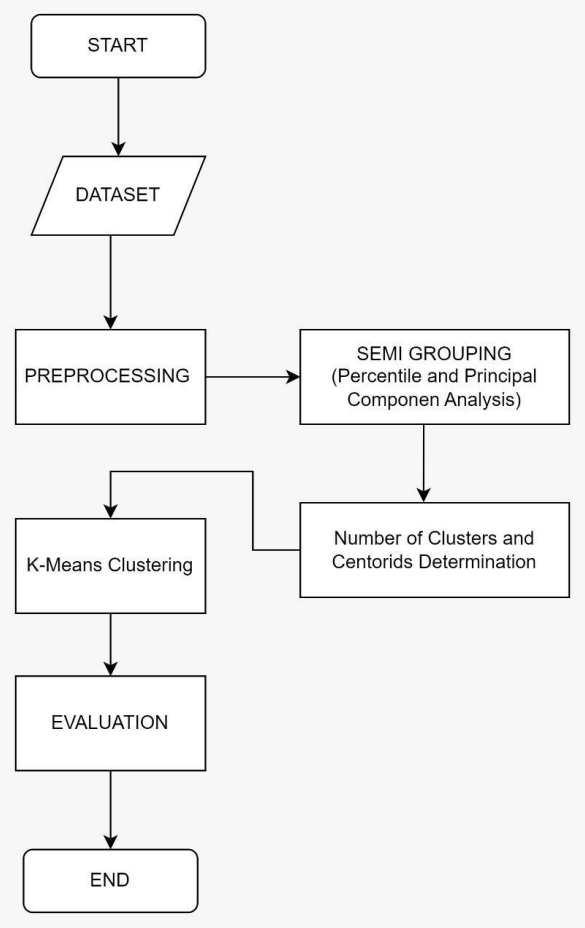


Figure1. Process Flow

The general K-Means procedure:

Algorithm 1 K-Means Clustering Algorithm

Input : A dataset D (D = {d_1, d_2, …, d_n}) and number of clusters K (K = K_1, K_2, …, K_n)

Output : K number of clusters

Procedure K-Means:

1. Take initial centroids, C= {c_1, c_2, …, c_k}
2. Assign each instance d_i to a cluster K_i by the closest distance
3. Calculate the new centroids by taking mean of each cluster
4. Repeat the above process until the centroids converges

Figure2. Procedure 1

Before using the following procedure, *note that the dataset has been sorted beforehand* :

Algorithm 2 Method of Initial Cluster Centroids

Input : A dataset D (D = {d_1, d_2, …, d_n}) and number of clusters K (K = K_1, K_2, …, K_n)
Note : D must be numeric

Output : centroids for K Clusters and cluster result

Procedure K-Means:

1. Applying principal component analysis, with 2 components (D must be scalled at first)
2. Applying percentile for grouping dataset into K equal parts based on 1st component
3. Take result of step 2 by index
4. Calculating the mean of each features of K equal parts
5. Take the mean (from step 4) as initial clusters centroids
6. Assign the initial clusters centroids to the K-Means clustering algorithm

Figure3. Procedure 2

As a reminder, to get the distance using the euclid principle, the following formula is used:

$$d = (i, j) = \sqrt{\Sigma_{k=1}^{n}[X_{ik} - X_{jk}]^2}$$

where i and j represent an object consisting of n components with finite dimensions.

For the implementation stage, the author uses the Python programming language which the author tries to develop from the principles that have been carried out by previous researchers. As for the accuracy of the method, it will be reviewed in relation to inter-cluster and intra-cluster. To find out how many best clusters are formed, the author measures using the silhouette coefficient, which is based on the average distance between points in the same cluster and in different clusters (nearest cluster);

$$s = \frac{b - a}{max(a, b)}$$

with:
a: average distance from a point/sample to all points in the cluster
b: average distance from a point/sample to all points in the nearest cluster

The following indicators are hereafter referred to as indices for model evaluation related to inter-cluster and intra-cluster:

1. Inter-cluster

For inter-cluster, the Davies-Bouldin Index (DBI) is used, measuring the average similarity between clusters so that it can be understood that if the DBI coefficient is low, which is close to 0, this means that the clusters formed are not similar, thus it can be said that if the DBI is low, the better the model to cluster the data. The DBI equation is given as follows:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} R_{ij}$$

where $R_{i,j}$ is the similarity measure between clusters, which also has similarity:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

with:

s_i :average distance between points/objects from cluster i and cluster i centre

d_{ij} : distance between cluster i and cluster j centres

2. Intra-cluster

For intra-cluster, the Calinski-Harabasz Index (CHI) is used, measuring the ratio of the SSE (cluster dispersion) of a cluster to the overall cluster so that it can be understood that if the value of this ratio is high, it can be said that the model is well clustering the data.

CHI takes the formula with the equation:

$$s = \frac{\mathrm{tr}(B_k)}{\mathrm{tr}(W_k)} \times \frac{n_E - k}{k - 1}$$

with

$$W_k = \sum_{q=1}^{k} \sum_{x \in C_q} (x - c_q)(x - c_q)^T$$

$$B_k = \sum_{q=1}^{k} n_q (c_q - c_E)(c_q - c_E)^T$$

Lihat[6].

C. Result and Discussion

In this case study, learning data is used which can be accessed at (https://www.kaggle.com/datasets/arjunbhasin2013/ccdata), where different principles are applied to this data to get the cluster results, then the computation time is compared and the similarity of the clusters formed. In the proposed procedure, there is preprocessing that is done first, namely sorting the column

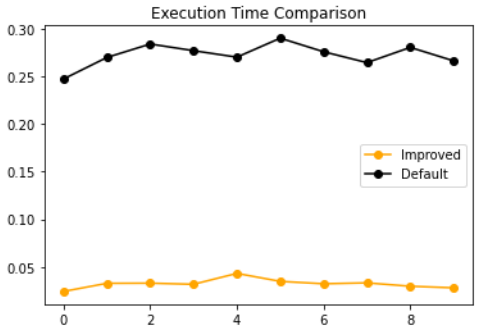
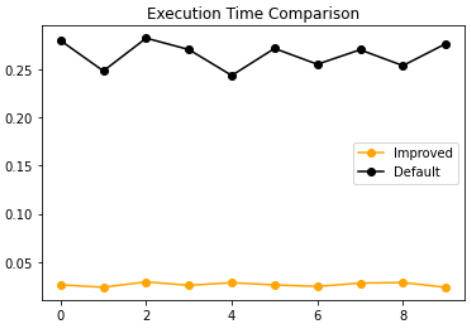
values of the data. The effect of this sorting can be seen in the following table : (Note that in this case, we use 3 clusters)

Table 1. Accuracy of Cluster Result

No	Treatment Type for Features	*Intra Cluster (Calinski Harabaz Index)*	*Inter Cluster (Davies Bouldin Index)*
1	Ordering - Grouping	7790.2	0.654
2	Grouping without Ordering	1546.53	1.603

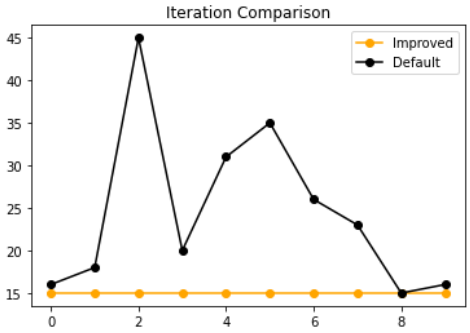
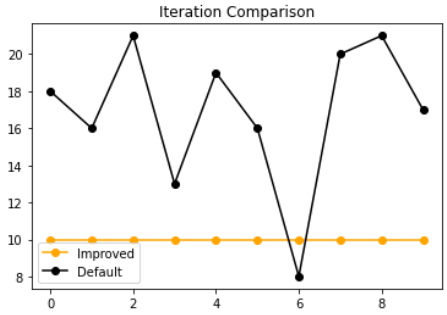
It can be seen that the results of ordering have a significant effect on cluster accuracy, where the level of similarity within the cluster (Calinski Harabasz Index) and similarity between clusters (Davies Bouldin Index) are high and low respectively, meaning that before clustering is implemented, preprocessing is like sorting the values will have an effect on the resulting clusters, then it will be seen from the computation time side until K-Means clustering is used.

Table 2. Time Comparison

No	Treatment Type	Execution Time
1.	Grouping without ordering	
2.	Ordering - grouping	

From Table 2 it can be seen that by using the ordering principle, computation time tends to be more stable and certainly more concise.

Table 3. Iteration Comparison

No	Treatment Type	Required Iterations
1.	Grouping without ordering	
2.	Ordering - grouping	

by ordering, fewer and consistent iterations are needed to get the final cluster, as a lesson from this case study it can be taken the point that by ordering better cluster results are obtained and the computation time and iterations required are more consistent.

D. Conclusion

From this case study it can be concluded that by preprocessing before implementing clustering algorithms such as scaling, sorting data, applying principal component analysis, and grouping with percentiles it can cut computation time both in terms of iterations and in terms of running time of the process, besides that it can also be reviewed the similarity of the clusters formed. In practice, how many clusters are needed depends on the domain that is the objective, for example in business, customer segmentation with 4 clusters is needed, so it would be better if several approaches were taken to get optimal cluster results and of course with a more effective running time.

E. Acknowledgment

Authors would like to thank the reviewers for their input in the preparation of this manuscript.

F. References

[1] A. Singh, A. Rana, and A. Yadav, "K-means with Three different Distance Metrics," *Int. J. Comput. Appl.*, vol. 67, no. 10, pp. 13–17, 2013.

[2] Seber, G., 1984, *Multivariate Observations*, John Wiley & Sons, New York.

[3] Yang, Q., C. Zhang, and S. Zhang, 2003. Data Preparation for Data Mining. *Applied Artificial Intelligence* 17:375-381

[4] Md. Zubair, MD. Asif Iqbal et al, An Improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling , *Annals of Data Science*, 2022.

[5] Afghani AS, Clustering with Euclidean Distance, Manhattan - Distance, Mahalanobis - Euclidean Distance, and Chebyshev Distance with Their Accuracy, *Indonesian Journal of Statistics and Its Applications , 2021.*

[6] Saitta, S., Raphael, B. and Smith, I.F.C. "A comprehensive validity index for clustering", *Intelligent Data Analysis, vol. 12, no 6,* 2008, pp. 529-548.

[7] Jiawei Han and Micheline Kamber, "Data Mining:Concepts and Techniques," *Morgan Kaufmann Publishers, August 2000. ISBN 1-55860-489-8.*

[8] Tan, Steinbach, Kumar Ghosh.The k-means algorithm -Notes.