The Application of the Minimum Forest Graph in Centroid Updating Stage of K-Means Algorithm

Achmad Maududie*, Wahyu Catur Wibowo**
* Universitas Jember
** Universitas Indonesia
*maududie@unej.ac.id, **wibowo@cs.ui.ac.id

ABSTRACT
K-Means is a well known algorithms of clusteing. It generates some groups based on degree of similarity. Simplicity of implementation, ease of interpretation, adaptability to sparse data, linear complexity, speed of convergence, and versatile in almost every aspect are noble characteristics of this algorithm. However, this algorithm is very sensitive on defining initial centroids process. Giving a bad initial centroid always produces a bad quality output. Due to this weakness, it is recommended to make some runs with different initial centroids and select the initial centroid that produces cluster with minimum error. However, this procedure is hard to achieve a satisfying result.

This paper introduces a new approach to minimize the initial centroid problem of K-Means algorithm. This approach focus on centroid updating stage in K-Means algorithm by applying minimum forest graph to produce better new centroids. Based on gain information and Dunn index values, this approach provided a better result than Forgy method when this approach tested on both well distributed and noisy dataset. Moreover, from the experiments with two dimentional data, the proposed approach produced consisten members of each cluster in every run, where it could not be found in Forgy method.

Keyword: K-Means, Minimum Forest Graph, Clustering

1. Introduction

Clustering is a very common term in many scientific disciplines, such as engineering, biology, medicine, and economics. Clustering became a major topic in 1960's and 1970's [1]. The goal of clustering is to discover a new set of categories from dataset based on their similarity or distance [2] without referring to any objects with prior identifiers [3]. Similar instances, which have close distance, are assigned in to one cluster while others are organized in to different groups and formally it is presented as $C = C_1, .., C_k$ of S[4]. This task is also called unsupervised learning or intrinsic classification [3].

In terms of its improvement, there are a lot of works that have been done to produce clustering algorithms. K-Means is one of famous algorithms based on partitioning clustering method, particularly on sum-of-squared error (SSE) criterion which is attempting to find a cluster with minimum distance of each instance [1]. It generates a single partition data for a single group of data that has high degree in similarity. K-Means generates *K* clusters that are represented by their centroids [4]. Each instance has a minimum distance to others in the same cluster but it is considered having as great as possible distance with other instances in different clusters.

Originally, it was proposed by several scientists in many forms and assumptions. Then it was investigated theoretical and algorithmic aspect by many researchers, such as Cox (1957), Fisher (1958), Bock (1970), Hartigan (1975), Diday et al. (1979), and Pollard (1982) [1]. In spite of having some advantages such as linear complexity, easy of interpretation, simplicity of implementation, speed of convergence and adaptability to sparse data [4], and versatile in almost every aspect [5], this method also has some weaknesses, such as very sensitive to initial centroids (center) [4], [5], [6]. The initial centroids drives the quality of clustering output. Due to this weakness, it is recommended to make some runs with different initial centroids and select the initial centroid that produces cluster with minimum error [6, p. 294]. Some researchers also tried to fix the initial centroid problem, such as Reddy et.al. used Voronoi Diagram [7], Cao et.al. used Neighborhood Mode [8], Shen and Meng used Small World Network [9], and Maududie and Wibowo used Minimum Forest Graph [10].

Actually, K-Means algorithm has iterative refinement mechanism in its process through rebuilding it centroids. Each new centroid is calculated as a means of all instances in each cluster [3], [4], [6]. However, in practical, this approach is not satisfying in the model refinement, particularly when the initial centroids are not well distributed. This paper introduces a new technique to enhance the refinement mechanism in K-Means

algorithm to reproduce a better model. In particular, this paper focuses on rebuilding a new centroid using minimum forest graph.

2. Research Method

2.1. The proposed approach

The simplest and most straightforward K-Means algorithm, namely Forgy method [3], comprises of 3 steps as follows [2], [4].

Input: ***S*** (instance set), ***K*** (number of cluster)
Output: clusters
1: **Initialize** ***K*** cluster centers (centroids).
While termination condition is not satisfied
2: Assign instances to the closest cluster center.
3: Update centroids based (***M***) on the assignment.

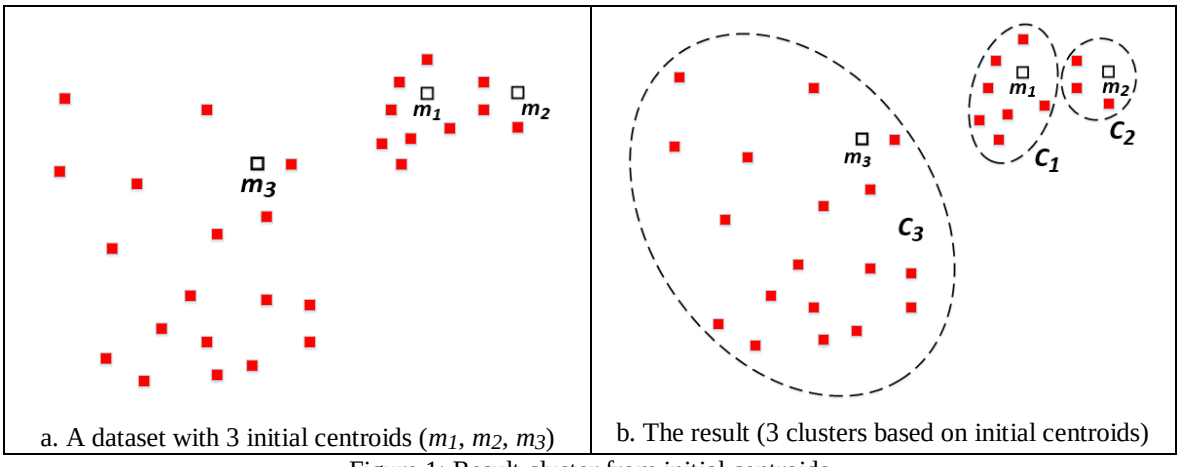
$$M_i = |C_i|^{-1} \sum_{x \in c_i} x$$

End

K-Means algorithm has iterative refinement (step 3) which is used to regenerate the centroids based on average of all instances in the same cluster label. We propose a new approach that attempt to improve this step to produce better new centroids based on minimum forest graph, which is also called nearest neighbor graph. In detail, the algorithm of this method is explained bellow.

Input: ***S*** (instance set), ***K*** (number of cluster)
Output: clusters
1: **Initialize** ***K*** cluster centers.
While termination condition is not satisfied
2: Assign instances to the closest cluster center.
3: Update centroids based (***M***) on the assignment.
a. Generate a set of seeds (***L***) in each cluster
b. Build minimum forest graph in each cluster
c. Use each center of tree (component forest) as candidate centroid (***M'***)
d. Generate true centroids based on existing candidate centroids
End

The first and the second step use the original Forgy method, i.e. select ***K*** points randomly from dataset (***S***) as initial centroid (cluster center) then assign each instance to the closest centroid. In this experiment, the distance between each instance is measured using cosine similarity. The result of this stage are ***K*** clusters where members of each cluster are highly depend on the initial centroids as illustrated in Figure 1.



a. A dataset with 3 initial centroids (m_1, m_2, m_3)

b. The result (3 clusters based on initial centroids)

Figure 1: Result cluster from initial centroids

On the 3rd step, the proposed approach introduces a new mechanism to update centroids which has 4 sub steps as follows.

a. *Generate a set of seeds* for each cluster ($\boldsymbol{L_i} = l_{i1}, l_{i2}, \ldots, l_{in}$) where maximum number of seeds ($\boldsymbol{n}$) per cluster is $2\boldsymbol{K}$ where $\boldsymbol{K}$ is number of clusters (see Figure 2a). The mechanism generate a set of seed by selecting $\boldsymbol{n}$ elements of each cluster member randomly. If the number of elements in one cluster less or equal than $\boldsymbol{n}$ then all elements within this cluster become seeds (cluster C_2).

b. *Create minimum forest graph* in every cluster from existing seeds based on their distance (see Figure 2b). One cluster may has more than one tree.

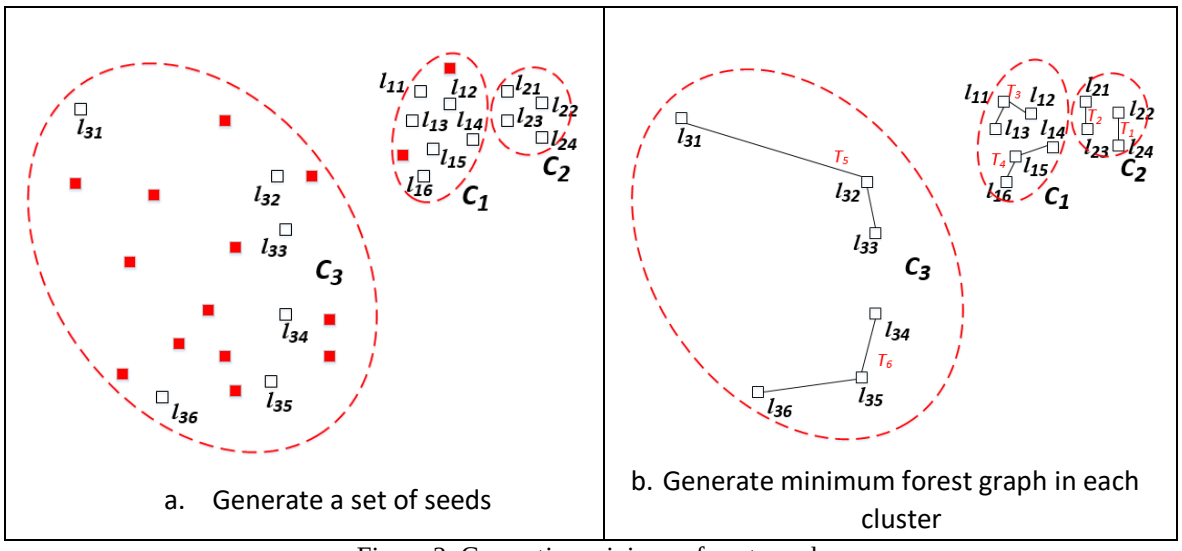


Figure 2: Generating minimum forest graph

c. *Use each center of tree as candidate centroid* ($\boldsymbol{M'} = m'_1, m'_2, \ldots, m'_z$) (see Figure 3a). The number of candidate centroids in one cluster may differ with other cluster because it depends on number of existing trees in each cluster. The center point is calculated as follows.

$$M'_i = |T_i|^{-1} \sum_{l \in T_i} l$$

d. *Generate true centroids* ($\boldsymbol{M} = m_1, m_2, \ldots, m_k$) using all the centroids in all clusters based on their distance (see Figure 3b). Therefore, it needs to calculate distance between one candidate centroid to the others. Merge the closest candidate centroids until the number of these centroids equal to the number of clusters ($\boldsymbol{K}$). When this condition is achieved, use this result as true centroid.

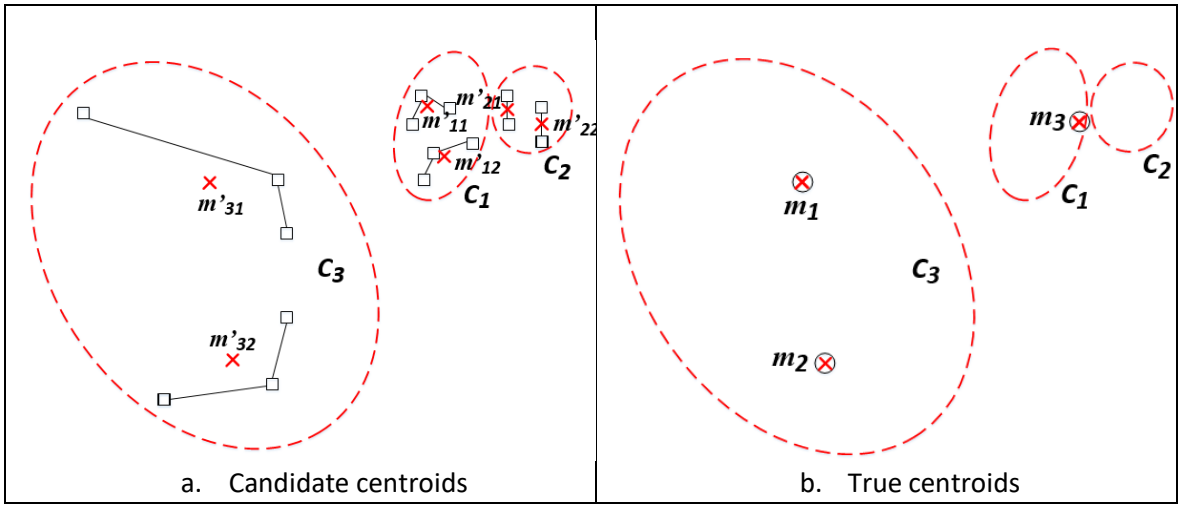


Figure 3: Generating true centroid based on minimum forest graph

The true centroid from step 3d is used to update centroids for the next process in K-Means algorithm until certain condition is accomplished. Figure 4 shows a comparison of assigning each instance to the closest centroids between initial centroids and updated centroids in the next process of K-Means.

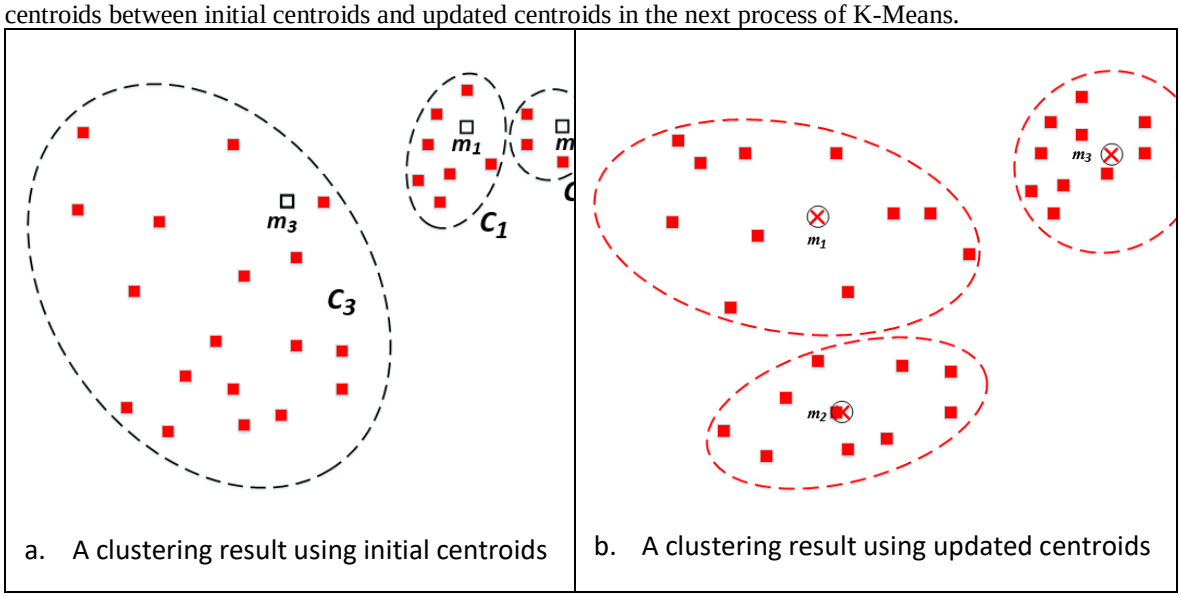


Figure 4: An Example of comparison of assigning instances to their closest centroid

In this paper, the quality of the proposed approach is evaluated using non-overlapping partitions evaluation schemes i.e. Dunn index and Information gain. Dunn index represents a ratio between the nearest distance of two objects in different cluster (d_{min}) and the farthest distance of two objects in the same cluster (the maximum diameter of cluster) (d_{max}). The larger the value of Dunn index indicates more compact and well-separated clusters [11] and could be expressed as follows [11].

$$D_{n_c=min_{1=1,\ldots,n_c}}\left\{min_{j=i+1,\ldots,n_c}\left(\frac{d(c_i,c_j)}{max_{k=1,\ldots,n_c}(diam(c_k))}\right)\right\}$$

or it could be simplify as bellows.

$$D = \frac{d_{min}}{d_{max}}$$

The next quality evaluator (Information gain) estimates the "amount of information" gained by clustering process that shows the degree of consistency between the distribution of elements and the partition of clusters which is described as follows [12].

$$Information\ Gain(X) = Info(D) - info_X(D)$$

Info(D) is the expected information to identify the class of an element in ***D*** which is calculated before partitioning occurs. If there is ***D*** dataset with *q* classes where *freq(**Cj**, D)* is the number of elements of the class ***Cj*** in ***D***, and |***D***|is number of total elements ***D***, then *Info(D)* is given by [12]:

$$Info(D) = -\sum_{j=1}^{q} \frac{freq(C_j, D)}{|D|} \times log_2\left(\frac{freq(C_j, D)}{|D|}\right).$$

When the partitioning process is applied and it gives *m* classes where |D| is the number of elements in D_i, the expected information, $Info_X(D)$, could be expressed bellows [12].

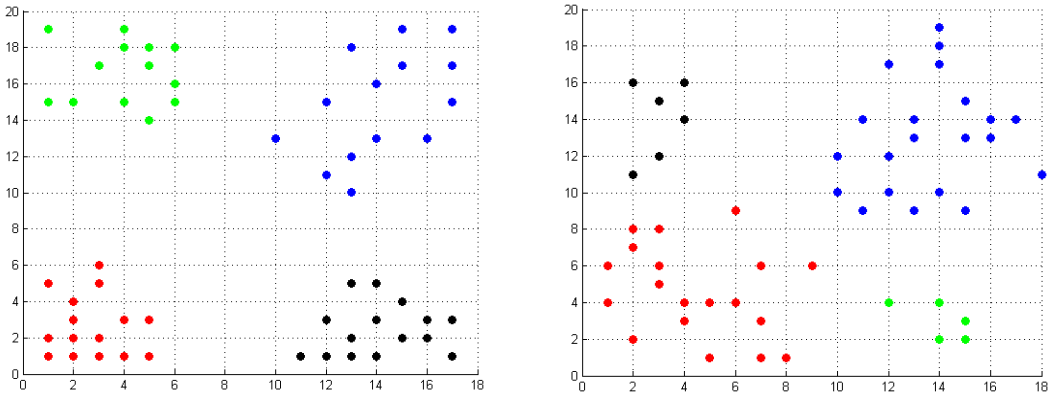
$$Info_X(D) = -\sum_{i=1}^{m} \frac{|D_i|}{|D|} \times Info(D_i).$$

2.2. Evaluation

In this paper, the proposed approach was evaluated using two dimensional (point) syntetic dataset. In this case we had two data testing that had four groups for each. The first data testing was made up of 60 data

points which were relatively well distributed (Figure 5a), while the second had 54 data points that were relatively noisy (Figure 5b). The results were compared with Forgy method to show the benefits, particularly the degree of consistency. In term of distance, the experiment used Euclidean distance to describe the dissimilarity of two data points which is calculated as bellows [13].

$$d(x,y) = \left(\sum_{j=1}^{d} (x_j, y_j)^2 \right)^{1/2}$$



a First data test b Second data test

Figure 5: The data test

3. Result and Analysis

As mentioned above, the proposed approach was evaluated using Dunn index and Information gain. The following tables illustrate the result clustering using the proposed and Forgy method with 10 runs for each.

Table 1: The result of clustering using Forgy method for the first data test

Run	Cluster	Points	Evaluation	
			Gain Information	Dunn Index
1	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16,17,18, 19, 20,21,22,23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
2	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
3	1	1, 2, 3, 4, 5, 6, 9, 11, 14	1.43	0.061
	2	7, 8, 10, 12, 13, 15		
	3	16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	22, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
4	1	1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15	1.50	0.061
	2	9, 10, 11, 13		
	3	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		

5	1	1, 2, 3, 4, 5, 6, 7, 8, 12, 14, 15, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45	1.50	0.116
	2	16, 17, 19, 21, 22, 27, 28		
	3	18, 20, 23, 24, 25, 26, 29, 30		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
6	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
7	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
8	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	1.50	0.086
	2	16,17,18, 19, 20,21,22,23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	3	46, 47, 48, 52, 54, 55, 56, 59, 60		
	4	49, 50, 51, 53, 57, 58		
9	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60	1.50	0.061
	2	16 ,17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 34, 36, 38, 39, 40 42, 43		
	4	33, 35, 37, 41, 44, 45		
10	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		

Table 2: The result of clustering using proposed method for the first data test

Run	Cluster	Points	Evaluation	
			Gain Information	Dunn Index
1	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16,17,18, 19, 20,21,22,23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
2	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
3	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
4	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454

	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
5	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
6	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
7	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
8	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
9	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		
10	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	2.00	0.454
	2	16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30		
	3	31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45		
	4	46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60		

Table 3: The result of clustering using Forgy method for the second data test

Run	Cluster	Points	Evaluation	
			Gain Information	Dunn Index
1	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 49, 50, 51, 52, 53, 54	1.35	0.063
	2	11, 42, 43, 44, 45, 46, 47, 48		
	3	20, 21, 22, 23, 24, 25, 27, 40		
	4	26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41		
2	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 53, 54	1.44	0.147
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 32, 33, 34, 38, 39, 40		
	3	30, 31, 35, 36, 37, 41, 49, 50, 51, 52		
	4	42, 43, 44, 45, 46, 47, 48		
3	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.55	0.185
	2	20, 21, 22, 23, 24, 25, 42, 43, 44, 45, 46, 47, 48		
	3	26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	4	49, 50, 51, 52, 53, 54		

4	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
5	1	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19	1.64	0.085
	2	5, 12, 13, 49, 50, 51, 52, 53, 54		
	3	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,40, 41		
	4	42, 43, 44,45, 46, 47, 48		
6	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
7	1	1, 2, 3, 5, 6, 7, 9, 10, 12, 13, 14, 15, 16, 17, 49, 50, 51, 52, 53, 54	1.25	0.071
	2	4, 8, 11, 18, 19, 42, 43, 44, 45, 46, 47, 48		
	3	20, 21, 22, 23, 24, 25, 27, 40		
	4	26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41		
8	1	1, 2, 3, 4, 5, 6, 7,8, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 49, 51, 52, 53, 54	1.30	0.063
	2	11, 42, 43, 44, 45, 46, 47, 48		
	3	20, 21, 22, 23, 24, 25, 26, 27, 40, 41, 50		
	4	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39		
9	1	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 14, 15, 16, 17, 18, 19	1.64	0.085
	2	5, 12, 13, 49, 50, 51, 52, 53, 54		
	3	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	4	42, 43, 44, 45, 46, 47, 48		
10	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		

Table 4: The result of clustering using proposed method for the second training dataset

Run	Cluster	Points	Evaluation	
			Gain Information	Dunn Index
1	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
2	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256

	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
3	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
4	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
5	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
6	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
7	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
8	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
9	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		
10	1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19	1.79	0.256
	2	20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		
	3	42, 43, 44, 45, 46, 47, 48		
	4	49, 50, 51, 52, 53, 54		

Table 1 and 3 show that the result of Forgy method is not consistent in each run, particularly when this method is applied to a dataset that has noisy data. It is not consistent because the initial centroid in every run is not consistent and the mechanism to update centroid is based on the average values of data point in every cluster. On the other hand, the proposed method gives a consistent result that is shown by their members of each cluster in every run (see Table 2 and 4). In addition, in terms of gain information, the proposed method conveys a better information in clustering than Forgy method. This achievement is indicated by the average values of gain information where the proposed method were 2.00 for the first dataset and 1.79 for the second dataset while Forgy method were 1.74 and 1.55 respectively. For the compactness point of view, the proposed method also gives more compact result that is denoted by the average of Dunn index, i.e.: 0.454 for the first dataset and 0.256 for the second dataset while Forgy method were 0.265 and 0.147 respectively.

4. Conclusion

Based on the conducted experiments, the enhancement of K-Mean algorithm through minimum forest graph provides a better solution on clustering process. At least there are three benefits of the proposed method to compare with Forgy method that are taken in to account, i.e.: consistency, conveying information, and compactness.

References

[1] H.-H. Bock, "Clustering Methods: A History of k-Means Algorithms," in *Selected Contributions in Data Analysis and Classification*, P. Brito, G. Cucumel, P. Bertrand, and F. de Carvalho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 161–172.

[2] P. S. Bradley and U. M. Fayyad, "Refining Initial Points for K-Means Clustering," in *ICML '98 Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 91–99.

[3] A. K. Jain, *Algorithms for clustering data*. Englewood Cliffs, N.J: Prentice Hall, 1988.

[4] L. Rokach, "A survey of Clustering Algorithms," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2010.

[5] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, Jan. 2013.

[6] J. MacQueen, "Some Methods for Classification and Analysis of MultiVariate Observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, vol. 1, pp. 281–297.

[7] D. Reddy and P. K. Jana, "Initialization for K-means Clustering using Voronoi Diagram," *Procedia Technol.*, vol. 4, pp. 395–400, 2012.

[8] F. Cao, J. Liang, and G. Jiang, "An initialization method for the K-Means algorithm using neighborhood model," *Comput. Math. Appl.*, vol. 58, no. 3, pp. 474–483, Aug. 2009.

[9] S. Shen and Z. Meng, "Optimization of Initial Centroids for K-Means Algorithm Based on Small World Network," in *Intelligent Information Processing VI*, vol. 385, Z. Shi, D. Leake, and S. Vadera, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 87–96.

[10] A. Maududie and W. C. Wibowo, "PERBAIKAN INISIALISASI K-MEANS MENGGUNAKAN GRAF HUTAN YANG MINIMUM," in *Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2014)*, Depok, Indonesia, 2014, pp. 8–15.

[11] M. Halkidi and M. Vazirgiannis, "Quality Assessment Approaches in Data Mining," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Boston, MA: Springer US, 2009, pp. 613–639.

[12] R. Quinian, *C4.5: programs for machine learning*. [S.l.]: Morgan Kaufmann, 1993.

[13] G. Gan, *Data clustering: theory, algorithms, and applications*. Philadelphia, Pa. : Alexandria, Va: SIAM, Society for Industrial and Applied Mathematics ; American Statistical Association, 2007.