Ali Ridho Barakbah, Yasushi Kiyoki, "A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation", IEEE Symposium on Computational Intelligence and Data Mining (CIDM) 2009, March 30-April 2, 2009, Nashville-Tennessee, USA.

A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation

Ali Ridho Barakbah and Yasushi Kiyoki, *Member, IEEE*

***Abstract*— Clustering performance of the K-means greatly relies upon the correctness of the initial centroids. Usually the initial centroids for the K-means clustering are determined randomly so that the determined centroids may reach the nearest local minima, not the global optimum. This paper proposes a new approach to optimizing the designation of initial centroids for K-means clustering. This approach is inspired by the thought process of determining a set of pillars' locations in order to make a stable house or building. We consider the pillars' placement which should be located as far as possible from each other to withstand against the pressure distribution of a roof, as identical to the number of centroids amongst the data distribution. Therefore, our proposed approach in this paper designates positions of initial centroids by using the farthest accumulated distance between them. First, the accumulated distance metric between all data points and their grand mean is created. The first initial centroid which has maximum accumulated distance metric is selected from the data points. The next initial centroids are designated by modifying the accumulated distance metric between each data point and all previous initial centroids, and then, a data point which has the maximum distance is selected as a new initial centroid. This iterative process is needed so that all the initial centroids are designated. This approach also has a mechanism to avoid outlier data being chosen as the initial centroids. The experimental results show effectiveness of the proposed algorithm for improving the clustering results of K-means clustering.**

I. Introduction

CLUSTERING is an effort to classify similar objects in the same groups. Cluster analysis constructs good cluster when the members of a cluster have a high degree of similarity of each other (internal homogeneity) and are not like members of other clusters (external homogeneity) [4, 9]. It means that the process to define a mapping $f:D \rightarrow C$ from some data $D=\{d_1,d_2,..,d_n\}$ to some clusters $C=\{c_1,c_2,..,c_n\}$ on similarity between d_i. The applications of clustering are diversely in many fields such as data mining, pattern recognition, image classification, biological sciences, marketing, city-planning, document retrievals, etc.

The most well known, widely used and fast methods for clustering is K-means algorithm developed by Mac Queen in 1967. The simplicity of K-means made this algorithm used in various fields. K-means algorithm is a partitioning clustering method that separates data into k mutually excessive groups. Through such the iterative partitioning, K-means clustering minimizes the sum of distance from each data to its clusters. K-means clustering is very popular because of its ability to cluster a kind of huge data, and also outliers, quickly and efficiently. It remains a basic framework for developing numerical or conceptual clustering systems because various possibilities of distance and prototype choice [2].

However, K-means clustering is very sensitive to the designated initial starting points as cluster centers. K-means clustering generates initial clusters randomly. If a randomly designated initial starting point closes to a final cluster center, then K-means clustering can find the final cluster center. It however is not always. If a designated initial point is far from the final cluster center, it will lead to incorrect clustering results [11]. Because of initial starting points generated randomly, K-means clustering does not guarantee the unique clustering results [13]. K-means clustering is difficult to reach global optimum, but only to one of local minima [7]. The better results of K-means clustering can be achieved after computing more than one times. However, it is difficult to decide the execution limit, which gives the best performance [17]. The cluster initialization algorithms for K-means tried to apply heuristic mechanisms to avoid the uncertainty of n times trial of K-means execution.

Several methods proposed to solve the cluster initialization for K-means clustering. A recursive method for initializing the means by running k clustering problems is discussed by Duda and Hart (1973). A variation of this method consists of taking the entire data into account and then randomly perturbing it k times [13]. Bradley and Fayyad (1998) proposed an algorithm that refines initial points by analyzing distribution of the data and probability of data density [3]. Penã et al. (1999) presented empirical comparison for four initialization methods for K-means clustering those are random, Forgy approach, Mac Queen approach, and Kaufman approach [5].

Our previous works have been dealing for K-means Optimization. Barakbah and Helen (2005) presented a new algorithm, called as Optimized K-means, that spreads the

This work was supported in part by the Japanese Ministry of Education, Culture, Sports, Science and Technology, and Keio University, under Grant Monbukagakusho scholarship and Mori Memorial Research Fund 2008 for Graduate Student Researcher Development Grant, Keio University.

A.R. Barakbah is a lecturer and head of Soft Computing Laboratory in Information Technology Department, Electronic Engineering Polytechnic Institute of Technology, Surabaya, Indonesia. Currently, he is a doctoral student at Graduate School of Media and Governance, Keio University, Japan, and belonging to Multi Database and Multimedia Database Laboratory, Keio University, Shonan Fujisawa Campus, Japan (corresponding author to provide phone: +81-80-3516-4979; fax: +81-466-486-093; e-mail: ridho@ mdbl.sfc.keio.ac.jp).

Y. Kiyoki is a professor of the Faculty of Environmental Information, Keio University, Japan. He is a supervisor of Multi Database and Multimedia Database Laboratory, Keio University, Shonan Fujisawa Campus, Japan (e-mail: kiyoki@sfc.keio.ac.jp).

978-1-4244-2765-9/09/$25.00 ©2009 IEEE

Ali Ridho Barakbah, Yasushi Kiyoki, "A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation",
IEEE Symposium on Computational Intelligence and Data Mining (CIDM) 2009, March 30-April 2, 2009, Nashville-Tennessee, USA.

initial centroids in the feature space so that the distances among them are as far as possible [14]. Barakbah et al. (2005) presented the optimization of initial starting points for K-means using Simulated Annealing [15]. Barakbah (2006) proposed a new algorithm to optimize the initial centroids for K-means by separately locating them as far as possible in data distribution [16]. By considering the computation of K-means optimization, Barakbah and Arai presented a fast algorithm for designating the initial centroids by embedding hierarchical algorithm into K-means clustering [17].

In this paper we propose a new approach for optimizing the initial centroids for K-means inspired by the designation of pillars' placement in a house or building. We consider the pillars which should be located as far as possible from each other to withstand against the pressure distribution of a roof, as the number of centroids amongst the data distribution. Therefore, our proposed approach determines the position of initial centroids by calculating the accumulated distance metric between each data and all previous centroids, and then, a data point which has the maximum distance will be selected. This approach is able to locate all centroids separately as far as possible between the initial centroids in the data distribution. This algorithm is also robust whereby it has a mechanism to avoid outlier data being chosen as the initial centroid. The detail description of the proposed approach will be described in forthcoming sections.

We organize this paper as follows. In Section 2, the K-means algorithm is described. Our proposed approach will be discussed in Section 3. Section 4 describes the validity measurements those are used in the experiments. Section 5 describes the experimental results using several benchmark datasets with several comparing algorithms, and then followed by concluding remarks in Section 6.

II. Basic Theory of K-Means

This section briefly explains the basic theory of K-means clustering for introducing our new approach to optimize the initial centroids for K-means in the next section. Let $A=\{a_i \mid i=1,\ldots,f\}$ be attributes of f-dimensional vectors and $X=\{x_i \mid i=1,\ldots,N\}$ be each data of A. The K-means clustering separates X into k partitions called clusters $S=\{s_i \mid i=1,\ldots,k\}$ where $M \in X$ is $M_i=\{m_{ij} \mid j=1,\ldots, n(s_i)\}$ as members of s_i, where $n(s_i)$ is number of members for s_i. Each cluster has cluster center of $C=\{c_i \mid i=1,\ldots,k\}$. K-means clustering algorithm can be described as follows:

1. Initiate its algorithm by generating random starting points of initial centroids C.
2. Calculate the distance d between X to cluster center C. Euclidean distance is commonly used to express the distance.
3. Separate x_i for $i=1..N$ into S in which it has minimum $d(x_i,C)$.
4. Determine the new cluster centers c_i for $i=1..k$ defined as:

$$c_i = \frac{1}{n_i} \sum_{j=1}^{n(s_i)} m_{ij} \in s_i \tag{1}$$

5. Go back to step 2 until all centroids are convergent.

The centroids can be said converged if their positions do not change in the iteration. It also may stop in the t iteration with a threshold ε [7] if those positions have been updated by the distance below ε:

$$\left| \frac{C^t - C^{t-1}}{C^t} \right| < \varepsilon \tag{2}$$

III. Proposed Algorithm

This section describes the basic concepts of the proposed algorithm, how to designate the initial centroids for optimizing K-means clustering, and an outlier detection mechanism. This section also contains a complexity analysis of the proposed algorithm and a sequential logical flow for the proposed approach.

A. Basic Concept

The K-means algorithm generates the initial centroids randomly and fails to consider a spread out placement of them spreading within the feature space. In this case, the initial centroids may be placed so close together that some become inconsequential. Because of this, the initial centroids generated by K-means may be trapped in the local optima. We propose in this paper a method of placing the initial centroids whereby each of them has a farthest accumulated distance between them.

The proposed algorithm in this paper is inspired by the thought process of determining a set of pillars' locations in order to make a stable house or building. Fig. 1 illustrates the locating of two, three, and four pillars, in order to withstand the pressure distributions of several different roof structures composed of discrete points. It is inspiring that by distributing the pillars as far as possible from each other within the pressure distribution of a roof, the pillars can withstand the roof's pressure and stabilize a house or

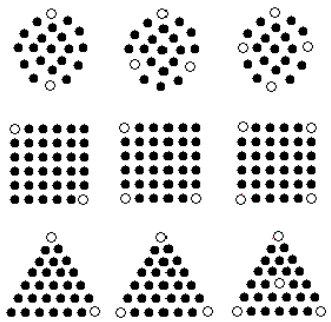

Fig. 1. Illustration of locating a set of pillars (white points) withstanding against different pressure distribution of roofs.

building.

this way, all centroids can be located as far as possible from

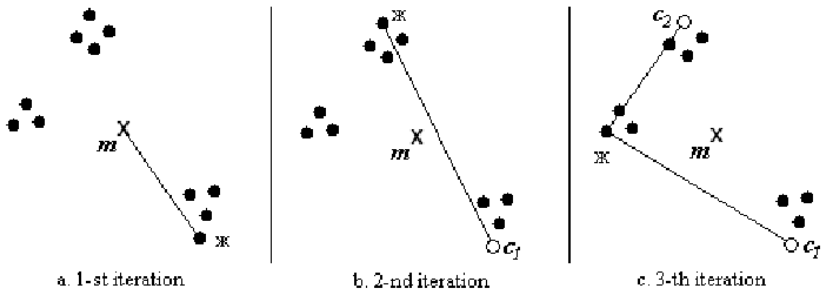

Fig. 2. Selection for several candidates of the initial centroids.

We consider the pillars which should be located as far as possible from each other to withstand against the pressure distribution of a roof, as number of centroids among the gravity weight of data distribution in the vector space. Therefore, our proposed approach in this paper designates positions of initial centroids in the farthest accumulated distance between them in the data distribution.

B. Determining Initial Centroids

First of all, the grand mean of data points is calculated as the gravity center of the data distribution. The distance metric D (let D^1 be D in this early step), is then created between each data point and the grand mean. A data point which has the highest distance in D^1 will be selected as the first candidate of the initial centroid ж. Fig. 2a illustrates m as the grand mean of data points and ж which is has the farthest distance to m is the candidate of the first initial centroid.

If ж is not an outlier, it will be promoted to the first initial centroid c_1. We then recalculate D (D^2 in this step), which is the distance metric between each data points and c_1. Starting from this step, we use the accumulated distance metric DM and assign D^2 to DM. This step which initiates the creation of DM is an improvement part of our previous work, MDC algorithm [16], that the construction of DM is started from D^1.

To select a candidate for the second initial centroid, the same mechanism is applied using DM instead of D. The data point with the highest distance of DM will be selected as the second initial centroid candidate ж, as shown in Fig. 2b. If ж is not classified as an outlier, it becomes c_2. To select a next ж for the candidate of the rest initial centroids, D^t (where t is the current iteration step) is recalculated between each data points and c_{t-1}. The D^t is then added to the accumulated distance metric DM ($DM \leftarrow DM + D^t$). This accumulation scheme can avoid the nearest data points to c_{t-1} being chosen as the candidate of the next initial centroid. It consequently can spread out the next initial centroids far away from the previous ones. The data points with the highest distance in DM will then be selected as ж, as shown in Fig. 2c.

If ж is not an outlier, it will become c_t. The iterative process guarantees that all initial centroids are designated. In each other within the data distribution. It follows, then, that our algorithm does not consider exhaustively search for an initial centroid near the gravity center of its group because that will later be handled by K-means after the end of the initial centroid generating algorithm. Our previous work proved [14] that the placement of an initial centroid in any location near its group will give the same result as the final centroid, as illustrated in Fig. 3 (the full red color point is the initial centroid and the red circle point is the final centroid after running K-means.

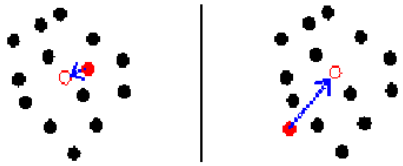

Fig. 3. An illustration for different placements of a centroid.

C. Outlier Detection Mechanism

To be selected as the initial centroids, the data point candidate must not be categorized as outlier. We identify an outlier by considering the number of neighbor points within the neighborhood boundary. Let n be the number of data points and k be the number of clusters, we set the number of neighbors $nmin$ by using a probabilistic parameter α to the average members of clusters n/k. For assuming the neighborhood boundary $nbdis$, we apply a threshold β to the

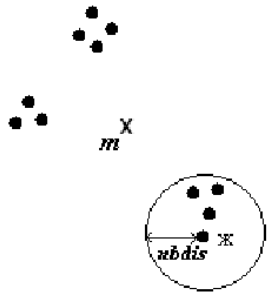

Fig. 4. An illustration of the neighborhood boundary.

highest distance in D^1, as shown in Fig. 4. In our case, we set α=0.25 and β=0.33, which means that a data point can be classified as an outlier if it has the number of neighbors lower than 25% of the average members of clusters within the neighborhood boundary which is 33% of the highest distance in D.

To get several data points to be the neighbors inside the boundary, it needs to calculate a distance metric between all data points and ж. However, it can utilize distance metric D for the next iteration step. In case shown in Fig. 4, the distance metric between all data points and ж can be acquired from D^2. If the number of neighbors inside the boundary is lower than *nmin*, ж will be classified as an outlier. When ж is considered as an outlier, it will be redetermined by selecting the second highest distance in D^1. This outlier detection mechanism is iteratively executed until ж can be considered as the initial centroid.

D.Algorithm Complexity

The time complexity of the algorithm proposed in this paper, is O((*k*+*h*+1) *n*), where *k* is number of clusters, *n* is number of data points, and *h* is number of outliers in the data set. This time complexity is little bit longer comparing to our previous work MDC algorithm which has O((*k*+1) *n*), since we involve an outlier mechanism whereas this mechanism was not covered by MDC. In a case of where there is no outliers in the data set, the time complexity can reach O((*k*+1) *n*), or equal to O((*t*+1) *n*), where *t* is the number of iterations, since selecting an initial centroid for each *k* takes one iteration. For worse case where there are number of outliers close to *n* (*h*≈*n*), the time complexity becomes ≈O(n^2). However, the outliers usually do not quantitatively dominate in a data set. For space complexity, this algorithm has O(2*n*), since it uses *DM* for storing the accumulated distance metric and *D* for storing the distance metric for each iteration, where $DM=\{dm_i \mid i=1,\ldots,n\}$ and $D=\{d_i \mid i=1,\ldots,n\}$.

E. Execution steps

Let $X=\{x_i \mid i=1,\ldots,n\}$ be data, *k* be number of clusters, $C=\{c_i \mid i=1,\ldots,k\}$ be initial centroids, $SX \subseteq X$ be identification for X which are already selected in the sequence of process, $DM=\{x_i \mid i=1,\ldots,n\}$ be accumulated distance metric, $D=\{x_i \mid i=1,\ldots,n\}$ be distance metric for each iteration, and *m* be the grand mean of X. The following execution steps of the proposed algorithm are described as:

1. Set $C=\varnothing$, $SX=\varnothing$, and $DM=[]$
2. Calculate $D \leftarrow \text{dis}(X,m)$
3. Set number of neighbors $nmin = \alpha \,.\, n / k$
4. Assign $dmax \leftarrow argmax(D)$
5. Set neighborhood boundary $nbdis = \beta \,.\, dmax$
6. Set i=1 as counter to determine the i-th initial centroid
7. $DM = DM + D$
8. Select ж $\leftarrow x_{argmax(DM)}$ as the candidate for i-th initial centroids
9. SX=SX ∪ ж
10. Set D as the distance metric between X to ж.
11. Set *no*←number of data points fulfilling $D \le nbdis$
12. Assign DM(ж)=0
13. If $no < nmin$, go to step 8
14. Assign $D(SX)$=0
15. $C = C \cup$ ж
16. $i = i + 1$
17. If $i \le k$, go back to step 7
18. Finish in which C is the solution as optimized initial centroids

IV. VALIDITY MEASUREMENTS

For performance analysis, we use several validity measures for clustering results; those are variance within cluster, variance between clusters, sum of squared error, standard deviation validity index, centroid proximity index, and execution time.

A. Variance Analysis

Variance constraint [9] can express the density of the clusters with the variance within cluster and the variance between clusters [6, 13]. The ideal cluster has minimum variance within cluster to express internal homogeneity and maximum variance between clusters to express external homogeneity [12]. Let $X=\{x_i \mid i=1,\ldots,N\}$ be data set, $S=\{s_i \mid i=1,\ldots,k\}$ be clustered X where $M \in X$ is $M_i=\{m_{ij} \mid j=1,\ldots,n(s_i)\}$ as members of s_i, variance within cluster can be defined as follows:

$$v_w = \frac{1}{N-k}\sum_{i=1}^{k}\left(n(s_i)-1\right)v_i^{\,2} \tag{3}$$

where N is number of data points, k is number of clusters, and n_i is number of members in i-th cluster, while v_i is given as:

$$v_i^{\,2} = \frac{1}{n(s_i)-1}\sum_{j=1}^{n(s_i)}\left(m_{ij}-\bar{s}_i\right)^2 \tag{4}$$

where m_j is members of i-th clusters.

Variance between clusters then can be defined as follows:

$$v_b = \frac{1}{k-1}\sum_{i=1}^{k} n(s_i)\left(\bar{s}_i-\bar{x}\right) \tag{5}$$

B. Sum of Squared Error

The most widely used criterion to quantify cluster homogeneity is the sum of squared error criterion [9]. It can be defined as:

$$SSE = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n(s_i)}\left\|m_{ij} - \bar{s}_i\right\|^2}{N} \quad (6)$$

C. Standard Deviation Validity Index

The standard deviation validity index is defined based on the concepts of the average scattering for clusters and total separation between clusters [8]. It consists of average scattering for clusters and total separation between clusters. The average scattering for clusters is defined as:

$$scat(k) = \frac{1}{k}\sum_{i=1}^{k}\frac{\left\|\sigma(s_i)\right\|}{\left\|\sigma(x)\right\|} \quad (7)$$

The definition of total scattering (separation) between clusters is given as follows:

$$dis(k) = \frac{D_{\max}}{D_{\min}}\sum_{i=1}^{k}\left(\sum_{j=1, j\neq i}^{k}\left\|s_i - s_j\right\|\right)^{-1} \quad (8)$$

where $D_{max} = \max(\|s_i - s_j\|)$ is the maximum distance between centroids, and $D_{min} = \min(\|s_i - s_j\|)$ is the minimum distance between centroids. The standard deviation validity index for k clusters can be defined as:

$$SDVI(k) = a \,.\, scat(k) + dis(k) \quad (9)$$

where a is a weighting factor equal to $dis(c_{max})$ where c_{max} is the maximum number of input clusters.

D. Centroid Proximity Index

We propose this kind of validity measurement, called centroid proximity index, to analyze the closeness of the final centroids of the clustering result to the centroids of the real data sets. It can be defined as:

$$CPI = \min\sum_{i=1}^{k}\left(\left\|c_i - r_i\right\|\right) \quad (10)$$

where c_i is i-th final centroid of clustering result and r_i is i-th real centroid of data set.

V. Experimental Results

To establish practical applicability of our proposed algorithm, we made a series of experiments and tested its performance on several benchmark datasets, mostly from UCI repository; those are Ruspini[1], Fossil [1], Iris, New Thyroid, Wine, Glass, Heart, and Ionosphere [18] datasets. For comparison purpose, we use the plain data of each data set without applying the normalization.

We conducted the performance comparison between the proposed algorithm and several approaches of initial centroids optimization for K-means; those are Forgy approach, Mac Queen approach, Kaufman approach, Refinement approach [3], MDC [16] and K-means using random initialization. For Forgy, Mac Queen and Refinement approach, because those approaches cannot give the unique clustering results, we made 10 times experiments and noticed their average results. We set α=0.25 and β=0.33 for our proposed algorithm to detect the outliers. In addition we also conducted the comparison between our proposed algorithm in this paper and several clustering algorithms those are hierarchical clustering (Single, Linkage, Linkage, and Average Linkage), and possibilistic clustering (Fuzzy C-Means). We run all programs under Matlab 6.5, with computer specifications: Intel Core2, 789 MHz, 2 GB RAM. We use built-in Matlab toolbox for Hierarchical algorithms. For Fuzzy C-Means, we used a program developed by David Corney (2000). Aside from that, we developed our own programs for the rest of algorithms.

Table 1-Table 8 shows the comparison results of validity measurements for each data set. Those tables perform that our proposed algorithm can optimize the designation of the initial centroids and then improve the precision of K-means clustering in all data sets and in most of validity measurements. Moreover, our proposed algorithm outperformed relatively in at least two validity measurements comparing to the other algorithms, and at least three validity measurements comparing to the other initial centroid optimization algorithms. However, the proposed algorithm got less performance in Glass and Heart data set. It was caused by inappropriately setting up the parameters of the proposed algorithm for outlier detection. If the parameters are adjusted too strictly, many data points will be considered as outlier, and it probably leads to discard the appropriate initial centroids. On the contrary, when the parameters are given broadly, the outliers can be selected as the initial centroids. In case of Glass data set, if we set more strictly β=0.1, the performance improves v_w and SSE respectively to 1.64 and 1.6, and outperforms the other algorithms. The opposite case related to Heart data set, if β is defined broader to 0.5, the performance becomes better for v_w=2047.25 and SSE=2024.5 which reaches same performance to MDC and outperforms to the rest of algorithms. For purpose of empirical study in this paper, we only simplified the parameter set-up and defined the same values in all data sets. However, the better way to set up these parameters for outlier detection is adjusted to characteristic of data distribution in the data set.

[1] built-in data object in both R and S-plus statistics packages.

Ali Ridho Barakbah, Yasushi Kiyoki, "A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation",
IEEE Symposium on Computational Intelligence and Data Mining (CIDM) 2009, March 30-April 2, 2009, Nashville-Tennessee, USA.

TABLE I
VALIDITY MEASUREMENT IN RUSPINI DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	284.50	78788.05	269.32	0.52	23.69
K-means with Forgy init.	388.01	77183.15	367.31	0.71	51.20
K-means with Mac Queen init.	436.15	77704.16	412.89	0.70	59.00
K-means with Kaufman init.	181.42	78190.89	171.75	0.29	0.00
K-means with Refinement init.	333.12	76886.69	315.35	0.46	34.28
K-means with MDC init.	181.42	78190.89	171.75	0.29	0.00
Single Linkage	181.42	78190.89	171.75	0.29	0.00
Complete Linkage	229.01	76572.48	216.79	0.32	9.47
Centroid Linkage	181.42	78190.89	171.75	0.29	0.00
Average Linkage	181.42	78190.89	171.75	0.29	0.00
Fuzzy C-Means	181.42	78190.89	171.75	0.29	0.00
K-means with our Pillar algorithm	181.42	78190.89	171.75	0.29	0.00

TABLE II
VALIDITY MEASUREMENT IN FOSSIL DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	42.92	6696.84	41.44	1.01	16.65
K-means with Forgy init.	41.45	6178.04	40.02	0.97	10.01
K-means with Mac Queen init.	43.41	6829.82	41.91	1.05	19.13
K-means with Kaufman init.	40.61	6011.59	39.21	0.89	9.52
K-means with Refinement init.	41.69	6212.71	40.25	0.97	9.12
K-means with MDC init.	45.46	7572.06	43.88	1.11	31.78
Single Linkage	50.28	7206.18	48.54	0.69	13.17
Complete Linkage	43.27	5894.69	41.78	1.25	5.54
Centroid Linkage	47.33	6713.86	45.70	0.78	9.27
Average Linkage	45.53	6498.85	43.96	0.85	6.68
Fuzzy C-Means	45.46	7572.06	43.89	1.11	31.50
K-means with our Pillar algorithm	40.58	6009.08	39.18	0.91	9.07

TABLE III
VALIDITY MEASUREMENT IN IRIS DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	0.59	181.47	0.58	4.81	3.83
K-means with Forgy init.	0.76	341.69	0.74	11.24	2.30
K-means with Mac Queen init.	0.62	318.32	0.61	7.80	1.25
K-means with Kaufman init.	0.54	302.23	0.53	5.85	0.51
K-means with Refinement init.	0.58	309.61	0.57	6.64	0.86
K-means with MDC init.	0.54	302.49	0.53	5.84	0.52
Single Linkage	0.97	317.98	0.95	4.34	2.54
Complete Linkage	0.61	300.84	0.60	5.22	0.86
Centroid Linkage	0.54	302.79	0.53	5.83	0.56
Average Linkage	0.54	302.79	0.53	5.83	0.56
Fuzzy C-Means	0.98	382.21	0.96	15.65	4.19
K-means with our Pillar algorithm	0.54	302.23	0.53	5.85	0.51

TABLE IV
VALIDITY MEASUREMENT IN NEW THYROID DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	136.40	19868.35	134.49	0.43	16.50
K-means with Forgy init.	136.85	22124.48	134.94	0.46	20.00
K-means with Mac Queen init.	136.41	20836.21	134.51	0.45	17.38
K-means with Kaufman init.	136.70	18581.93	134.80	0.44	16.62
K-means with Refinement init.	137.19	26366.74	135.28	0.48	26.30
K-means with MDC init.	137.73	27970.57	135.81	0.48	29.71
Single Linkage	278.14	119041.95	274.26	0.81	95.97
Complete Linkage	166.84	42480.35	164.51	0.40	54.85
Centroid Linkage	222.60	78800.70	219.49	0.17	87.38
Average Linkage	213.55	40171.26	210.57	0.27	61.19
Fuzzy C-Means	136.24	18621.09	134.34	0.43	13.73
K-means with our Pillar algorithm	136.21	18629.12	134.31	0.42	13.90

Ali Ridho Barakbah, Yasushi Kiyoki, "A Pillar Algorithm for K-Means Optimization by Distance Maximization for Initial Centroid Designation", IEEE Symposium on Computational Intelligence and Data Mining (CIDM) 2009, March 30-April 2, 2009, Nashville-Tennessee, USA.

TABLE V
VALIDITY MEASUREMENT IN WINE DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	14898.68	9702596.49	14647.58	0.03	449.30
K-means with Forgy init.	13847.22	8228689.09	13613.84	0.04	286.00
K-means with Mac Queen init.	14450.95	8933587.32	14207.40	0.03	365.34
K-means with Kaufman init.	13546.80	7807572.69	13318.48	0.04	239.35
K-means with Refinement init.	14597.18	9220246.43	14351.16	0.03	396.97
K-means with MDC init.	15048.89	9913154.68	14795.25	0.03	472.63
Single Linkage	78592.92	29148962.94	77268.32	0.10	1634.98
Complete Linkage	14062.37	8119822.76	13825.36	0.03	279.94
Centroid Linkage	22298.57	17017412.83	21922.75	0.03	989.86
Average Linkage	22298.57	17017412.83	21922.75	0.03	989.86
Fuzzy C-Means	13579.61	7829807.43	13350.74	0.04	248.09
K-means with our Pillar algorithm	13546.80	7807572.69	13318.48	0.04	239.35

TABLE VI
VALIDITY MEASUREMENT IN GLASS DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	1.74	279.49	1.69	54.97	14.14
K-means with Forgy init.	2.11	243.28	2.05	66.83	11.62
K-means with Mac Queen init.	1.80	255.57	1.75	67.99	12.93
K-means with Kaufman init.	1.86	283.68	1.81	47.50	13.54
K-means with Refinement init.	2.06	434.16	2.00	298.05	20.74
K-means with MDC init.	2.07	529.32	2.01	18.77	21.49
Single Linkage	5.18	407.03	5.03	16.79	31.59
Complete Linkage	1.96	356.65	1.90	23.74	18.28
Centroid Linkage	4.42	409.72	4.29	16.79	29.86
Average Linkage	4.25	313.30	4.13	16.37	26.28
Fuzzy C-Means	2.05	318.98	1.99	40.87	12.45
K-means with our Pillar algorithm	2.06	303.50	2.00	34.16	12.19

TABLE VII
VALIDITY MEASUREMENT IN HEART DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	2670.32	2481847.84	2640.65	0.01	230.75
K-means with Forgy init.	2116.84	497918.34	2093.32	0.02	85.95
K-means with Mac Queen init.	2047.33	250516.93	2024.59	0.02	67.89
K-means with Kaufman init.	2049.01	246558.31	2026.25	0.02	67.60
K-means with Refinement init.	2462.52	1738477.00	2435.16	0.02	176.49
K-means with MDC init.	2047.25	251126.45	2024.50	0.02	67.93
Single Linkage	2844.36	4584128.45	2812.75	0.01	322.74
Complete Linkage	2592.41	769208.35	2563.61	0.02	129.84
Centroid Linkage	2739.35	2730213.34	2708.92	0.01	248.88
Average Linkage	2739.35	2730213.34	2708.92	0.01	248.88
Fuzzy C-Means	2047.69	249139.02	2024.94	0.02	67.80
K-means with our Pillar algorithm	2049.01	246558.31	2026.25	0.02	67.60

TABLE VIII
VALIDITY MEASUREMENT IN IONOSPHERE DATA SET

Algorithm	v_w	v_b	*SSE*	*SDVI*	*CPI*
K-means with random init.	6.93	829.40	6.89	0.49	2.39
K-means with Forgy init.	6.93	829.00	6.89	0.49	2.38
K-means with Mac Queen init.	8.06	1828.71	8.01	0.42	4.09
K-means with Kaufman init.	6.93	828.60	6.89	0.49	2.38
K-means with Refinement init.	6.93	829.40	6.89	0.49	2.39
K-means with MDC init.	6.93	829.40	6.89	0.49	2.39
Single Linkage	9.19	3205.77	9.14	0.33	6.26
Complete Linkage	8.83	351.26	8.78	0.66	1.68
Centroid Linkage	9.19	3205.77	9.14	0.33	6.26
Average Linkage	9.19	3205.77	9.14	0.33	6.26
Fuzzy C-Means	6.93	830.08	6.89	0.49	2.39
K-means with our Pillar algorithm	6.93	828.60	6.89	0.49	2.38

TABLE IX
COMPARISON OF EXECUTION TIMES (IN SECOND) FOR INITIAL CENTROIDS OPTIMIZATION ALGORITHMS OF K-MEANS FOR EACH DATA SET

Algorithm	Ruspini	Fossil	Iris	New Thyroid	Wine	Glass	Heart	Ionosphere
K-means with random init.	0.01	0.01	0.00	0.02	0.01	0.04	0.00	0.00
K-means with Forgy init.	0.13	0.13	0.20	0.28	0.24	0.55	0.16	0.32
K-means with Mac Queen init.	0.14	0.12	0.21	0.29	0.25	0.56	0.18	0.36
K-means with Kaufman init.	21.05	16.31	49.05	101.19	69.72	384.34	29.20	115.70
K-means with Refinement init.	3.98	3.46	12.84	50.96	41.22	66.79	12.50	69.23
K-means with MDC init.	0.19	0.16	0.23	0.31	0.27	0.59	0.19	0.34
K-means with our Pillar algorithm	0.19	0.36	0.55	2.33	0.42	2.08	0.42	27.05

Table 9 shows the comparison of execution times for optimization algorithms of K-means' initial centroids for each data set. The execution time of our proposed algorithm in all data sets is longer than MDC because it involved the outlier detection mechanism. However, our proposed algorithm is constantly better than both the Kaufman and Refinement approach. Compared to the Forgy and Mac Queen approach, it seems that the execution time of our proposed algorithm is worse. But, the performance of computational time for those two comparing algorithms is computed from an average of 10 executions. In a real world situation, to get the best performance for either, it needs to run more than one execution to get a non-unique clustering result. In this kind of situation, our proposed algorithm is sufficient with only one execution time due to its uniqueness of clustering result.

VI. CONCLUSION

A new algorithm for optimization of K-means clustering, inspired by pillar placement is proposed in this paper. It designates the initial centroids' positions by calculating the accumulated distance metric between each data point and all previous centroids, and then selects data points which have the maximum distance as new initial centroids. Our algorithm distributes all initial centroids according to the maximum accumulated distance metric. The algorithm also introduces a mechanism for detecting outliers. Several experiments involving eight benchmark data sets with five validity measurements and execution time were conducted. The experimental results show that our proposed algorithm is able to optimize the selection of initial centroids and improve the K-means precision in all data sets and in most of the validity measurements. Moreover, our proposed algorithm outperformed the other algorithms in at least two validity measurements, and the other initial centroid optimization algorithms in at least three validity measurements. However, inappropriate parameter set-up in the proposed algorithm for outlier detection may lead to reduced performance. Adjusting to the characteristics of the data distribution in the data set is needed in order to set-up the appropriate parameters for the outlier detection mechanism. Our proposed algorithm's execution time shows better performance than the other initial centroid optimization algorithms, except MDC.

REFERENCES

[1] C. Yi-tsuu, "Interactive Pattern Recognition," Marcel Dekker Inc., New York and Basel, 1978.
[2] H. Ralambondrainy, "A conceptual version of the K-means algorithm," *Pattern Recognition Lett.*, 16, 1147-1157, 1995.
[3] P.S. Bradley, U.M Fayyad, "Refining initial points for K-means clustering," *Proc. 15th Internat. Conf. on Machine Learning* (ICML'98), 1998.
[4] G.A. Growe, "Comparing Algorithms and Clustering Data: Components of The Data Mining Process," thesis, department of Computer Science and Information Systems, Grand Valley State University, 1999.
[5] J.M. Penã, J.A. Lozano, P. Larrañaga, "An empirical comparison of the initilization methods for the K-means algorithm," *Pattern Recognition Lett.,* 20, 1027-1040, 1999.
[6] S. Ray, R.H. Turi, "Determination of number of clusters in K-means clustering and application in colthe image segmentation," *Proc. 4th ICAPRDT*, pp.137-143, 1999.
[7] B. Kövesi, J.M. Boucher, S. Saoudi, "Stochastic K-means algorithm for vector quantization," *Pattern Recognition Lett.*, 22, 603-610, 2001.
[8] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "Clustering algorithms and validity measures," *Proc. 13th International Conference on Scientific and Statistical Database Management* (SSDBM'01), 2001.
[9] C.J. Veenman, M.J.T. Reinders, E. Backer, "A maximum variance cluster algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1273-1280, 2002.
[10] V.E. Castro, "Why so many clustering algorithms-a position paper," *ACM SIGKDD Explorations Newsletter*, Vol. 4, Issue 1, pp. 65-75, 2002.
[11] Y.M. Cheung, "k*-Means: A new generalized k-means clustering algorithm," *Pattern Recognition Lett.*, 24, 2883-2893, 2003.
[12] A. R. Barakbah, K. Arai, "Identifying moving variance to make automatic clustering for normal dataset," *Proc. IECI Japan Workshop 2004* (IJW 2004), Musashi Institute of Technology, Tokyo, 2004.
[13] S.S. Khan, A. Ahmad, "Cluster center initialization algorithm for K-means clustering," *Pattern Recognition Lett.*, 25, 1293-1302, 2004.
[14] A.R. Barakbah, A. Helen, "Optimized K-means: an algorithm of initial centroids optimization for K-means," *Proc. Soft Computing, Intelligent System, and Information Technology* (SIIT) 2005, pp.2-63-66, Petra Christian University, Surabaya, 2005.
[15] A.R. Barakbah, A. Fariza, Y. Setiowati, "Optimization of Initial Centroids for K-means using Simulated Annealing," *Proc. Industrial Electronics Seminar* (IES) 2005, pp.286-289, Electronic Engineering Polytechnic Institute of Surabaya-ITS, Surabaya, 2005.
[16] A.R. Barakbah, "A new algorithm for optimization of K-means clustering with determining maximum distance between centroids," *Proc. Industrial Electronics Seminar* (IES) 2006, pp.240-244, Electronic Engineering Polytechnic Institute of Surabaya-ITS, Surabaya, 2006.
[17] A.R. Barakbah, K. Arai, "Hierarchical K-means: an algorithm for centroids initialization for K-means," *Reports of the Faculty of Science and Engineering*, Saga University, Japan, Vol. 36, No. 1, 2007.
[18] UCI Repository (http://www.sgi.com/tech/mlc/db/)