British Journal of Computer, Networking and Information Technology
ISSN: 2689-5315
Volume 4, Issue 1, 2021 (pp. 28-41)
www.abjournals.org

NEW K-MEANS CLUSTERING METHOD USING MINKOWSKI'S DISTANCE AS ITS METRIC

Eric U. Oti[1*], Michael O. Olusola[2], Oberhiri-Orumah Godwin[3] and Chike H. Nwankwo[2]

[1]Department of Statistics, Federal Polytechnic, Ekowe Bayelsa State

[2]Department of Statistics, Nnamdi Azikiwe University, Awka Anambra State

[3]Federal Polytechnic Library, Federal Polytechnic, Ekowe Bayelsa State

*Corresponding Author: eluchcollections@gmail.com (Tel: +2348037979262)

Cite this article:

Eric U.O., Michael O.O., Oberhiri-Orumah G., Chike H. N. (2021), New K-means Clustering Method Using Minkowski's Distance as its Metric. British Journal of Computer, Networking and Information Technology 4(1), 28-41. DOI: 10.52589/BJCNIT-XEPSJBWX.

Manuscript History

Received: 28 May 2021

Accepted: 25 June 2021

Published: 6 July 2021

ABSTRACT: *Cluster analysis is an unsupervised learning method that classifies data points, usually multidimensional into groups (called clusters) such that members of one cluster are more similar (in some sense) to each other than those in other clusters. In this paper, we propose a new k-means clustering method that uses Minkowski's distance as its metric in a normed vector space which is the generalization of both the Euclidean distance and the Manhattan distance. The k-means clustering methods discussed in this paper are Forgy's method, Lloyd's method, MacQueen's method, Hartigan and Wong's method, Likas' method and Faber's method which uses the usual Euclidean distance. It was observed that the new k-means clustering method performed favourably in comparison with the existing methods in terms of minimization of the total intra-cluster variance using simulated data and real-life data sets.*

KEYWORDS: Clustering, Cluster Centres, Euclidean's Distance, Minimum Distance Rule, Minkowski's Distance.

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

INTRODUCTION

Clustering as a mechanism is applied to a wide range of disciplines like anthropology, bioinformatics, biology, computer science, data mining, geography, marketing, psychology, statistics (Anderberg, 1973; Everitt et al., 2011) and have a common goal which is to partition a given set of data points into clusters such that similar points are classified to the same cluster, whereas dissimilar ones are not (Forgy, 1965; MacQueen, 1967; Hartigan, 1975; Lloyd, 1982). The most popular formulation in clustering is k-means which purpose is to either minimize the total intra-cluster variance or to maximize the expected similarity between data points and their associated cluster centres (Slonim et al., 2013). Johnson and Wichern (2002) stated that clusters of data points should exhibit high internal (within-clusters) homogeneity and high external (between-clusters) heterogeneity; clustering is done based on similarities or distances and it is divided into two main groups which are based on the structure of their output namely: hierarchical and non-hierarchical clustering methods. Hierarchical clustering is a method of cluster analysis that seeks to build a hierarchy of clusters. The clusters are merged (agglomerative methods) or split (divisive methods) step-by-step based on the applied similarity measure. The results of a hierarchical clustering method entail that agglomerative and divisive methods can be displayed graphically using a tree-like diagram known as a dendrogram. While non-hierarchical or partitioning clustering methods partition the data object set into clusters where every pair of object clusters is either distinct (non-overlapping) or has some members in common (overlapping), partitioning clustering begins with a starting cluster partition which is iteratively improved until a locally optimal partition is reached. Amongst the partitioning clustering methods, the k-means method is the most popularly and commonly used in practice.

The purpose of this paper is to propose a new k-means clustering method that uses Minkowski's distance as its metric when calculating between each data points and the cluster centroids.

The rest of this paper is organized as follows: Section 2 discusses the methods used in this paper. In section 3, the experimental results of the simulated data and the real-life data is shown and discussed. Section 4 is the conclusion of the paper.

METHODOLOGY

Several k-means clustering methods aim to classify points or objects to be analyzed into well-separated groups (clusters). Six k-means clustering methods will be discussed in this paper and the proposed method is a batch algorithm that uses the Minkowski distance instead of the usual Euclidean distance. Minkowski's distance has some advantages over the Euclidean distance that is used in the k-means clustering methods, and it is based on the fact that it is easy to compute and allows scalable solutions of other problems such as clustering and indexing (Gunopulos and Das, 2001). The rationale behind this developed method is based on the assumption that an optimal clustering solution with k clusters can be obtained through local search. To be able to use any of the six methods, the number of clusters present in the data need to be known; multiple runs or trials will be necessary to find the best number of clusters. There is no best method, as the tendency of generating global optimum depends on the characteristics of the data set, size, and the number of variables in the cases. The k-means clustering methods have two phases of iteration namely: the assignment or initialization phase which involves an iterative process where each data point is assigned to its nearest centroid using any metric of choice; the next is the centroid update phase, where clusters centroids are updated given the partition obtained by the previous phase. The iterative process stops when no data point change clusters or some maximum number of iterations is reached.

Forgy's Method

Forgy (1965) proposed a batch algorithm which is seldom referred to as the traditional k-means algorithm. The algorithm is an offline centroid clustering model which is based on the minimization of the average squared Euclidean distance between the data points and the cluster's centre known as centroid. A centroid is the centre of a geometric object and it is seen as a generalization of the mean. A batch algorithm is an algorithm where a transformative step is applied to all data-point (case) at once, where c is the cluster centre in the Euclidean distance and x is the case, i is the dimension of $x(or\ c)$ being compared and k is the total number of dimensions. That is,

$$d_{euc} = \sqrt{\sum_{i=1}^{k} \quad (c_i - x_i)^2} \tag{1}$$

being the most common distance. Forgy's method starts with the choosing of k instance or initialization of data set uniformly at random and assigns the rest of the data points to the closest cluster (Peña et al, 1999). This method is very applicable because of its simplicity and high-speed intensity. It also treats the data set as a continuous distribution. Given the data set $\{x_1, x_2, \ldots, x_n\} \epsilon R^d$, where R^d is the real d-dimensional data space (or the Euclidean d-dimensional data space), the algorithm tries to find a set of k cluster centres, $c = \{c_1, c_2, \ldots, c_k\} \epsilon R^d$. The error function for a continuous distribution is defined as

$$E = \sum_{i=1}^{k} \quad \int \qquad f(x) d(c_i, x_i) dx \tag{2}$$

In the above equation, $f(x)$ is the probability density function at x and $d(c_i, x_i)$ is the distance function. We note that if the probability density function is not given (or known), then it has to be deduced (generated) from the given data. Though the k-means algorithm converges to a local optimum, the limit point depends on the initial points. Hence, it is appropriate to start with a reasonable initial partition in order to realize a high-quality clustering solution. However, there is no efficient and universal technique for obtaining such initial partitions theoretically.

Forgy's method has a major drawback, the possibility of choosing an outlier as an initial cluster centre, in such a case, it is possible that no other data point is assigned to it, and hence the cluster with the outlier as its centre remains singleton. Also, there is no mechanism to avoid choosing data points that are very close to each other cluster centres.

Algorithm.1: The Forgy's (Traditional) Algorithm.

1. Begin with any desired initial configuration. Go to step 2 if beginning with a set of seed points; go to step 3 if beginning with a partition of the data units.
2. Assign each data unit to the cluster with the nearest seed point. The seed points remain fixed for a full cycle through the entire data set.

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

3. Compute new seed points as the centroids of the cluster of the data units.

4. Repeat step 2 and 3 until the process converges; that is, continue until no data units change their cluster membership at step 2.

Lloyd's Method

Lloyd (1982) proposed a method that is widely known as the standard k-means algorithm; it is also a batch algorithm that is based on the minimization of the average squared Euclidean distance between the data items and the cluster centres like Forgy's method. The dissimilarity between the Lloyd algorithm and the Forgy algorithm is that the Lloyd algorithm treats the data set as a discrete distribution while the Forgy algorithm treats the data set as a continuous distribution. While the similarity between them is that they have the same procedure, the error function for a discrete distribution is defined as

$$E = \sum_{i=1}^{k} \quad \sum_{j=1}^{n} \quad f(x)d(c_i, x_i) \qquad (3)$$

In Equation (3) above, $d(c_i, x_i)$ is the distance function of the data point x_i and cluster centre c_i. The first step of the algorithm begins with choosing the number of clusters k and its initial centroids or cluster centres. It could be done by either using k random observations or from the k observations that are the farthest from one another in the data space. Initialization of the centroids occurs only once, and once the initial centroids have been chosen, iterations are done on the following two steps. First, the data set is assigned to cluster centroids (centres), using any of the distance metrics. All cases assigned to a centroid are said to be part of the centroids subspace c (R^d) (Morissette and Chartier, 2013). Second, update the value of the centroid by using the mean of the data points (cases) assigned to the centroid.

Algorithm 2: The Lloyd's (Standard) Algorithm.

1. Choose k data objects representing the cluster centroids.

2. Assign each data object of the entire data set to the cluster having the closest centroid.

3. Compute a new centroid for each cluster by averaging the data observations belonging to the cluster.

4. If at least one of the centroids has changed, go to step 2, otherwise, go to step 5

5. Output the clusters.

MacQueen's Method

MacQueen (1967) proposed MacQueen's algorithm, and it is often referred to as a basic k-means algorithm which is an online (or incremental) algorithm. MacQueen's method is similar to Forgy's and Lloyd's Methods, but the main difference is that the centroids are updated by re-calculating the points (cases) any time it is moved. Once the initial centroids have been chosen in the same way as Lloyd's algorithm, the iterations follow: For each case (x_i) in turn, after arbitrarily partitioning of points (items) into clusters, we compute the coordinates ($\underline{x}_i'^{s}$) of the cluster centroid (mean), likewise the Euclidean distance is computed for each point from the group centroids and reassign each point to the nearest group. If a point is moved from its initial position, the cluster centroid must be recalculated or updated before computing the squared distances.

If the centroid of a case belongs to the nearest subspace, no change is made. If another centroid is closest to the subspace, the case is re-assigned to the other centroid and the centroids for both the old and new subspaces (centres) are recalculated as the mean of the cases. When we see that each point is currently assigned to the clusters with the nearest centroid, the process stops.

Algorithm 3: The MacQueen's (Basic) Algorithm.

1. Choose k points as initial cluster centroids.

2. Assign each object to the cluster that has the closest centroid.

3. When all objects have been assigned, re-compile the positions of the k centroids.

4. If at least there is a change in one of the centroids, repeat step 2 and 3, otherwise go to step 5.

5. Output the clusters.

Hartigan and Wong's Method

Hartigan and Wong (1979) proposed a non-Lloyd heuristic method known as a conventional k-means algorithm that updates centres considering each point, rather than after each pass over the entire data set.

The algorithm searches for the partition of data space with the locally optimal within-cluster sum of squares error (SSE), which means that it may assign a case to another subspace, even if it currently belongs to the subspace of the closest centroid; if doing so minimizes the total within-cluster sum of squares (Morissette and Chartier, 2013). The initialization of the cluster centres is done in the same way as that of Lloyd's and Forgy's algorithms. The points (cases) are designated (assigned or allotted) to the centroid nearest to them and the centroids are calculated as the mean of the designated data points. The iterative steps are as follows:

Step 1. For each point I$(I = 1, \dots, M)$, find its closest and second closest cluster centres, $IC1(I)$ and $IC2(I)$, respectively. Assign point I to cluster $IC1(I)$.

Step 2. Update the cluster centres to be the average of the points contained within them.

Step 3. Initially, all clusters belong to the live set (specified number of k).

Step 4. This is the optimal transfer (OPTRA) stage: Consider each point I $(I = 1, 2, \dots, M)$ in turn. If cluster L $(L = 1, 2, \dots, K)$ is updated in the last quick-transfer (QTRAN) stage, then the cluster belongs to the live set throughout this stage. Otherwise, at each step, it is not in the live set if it has not been updated in the last M optimal-transfer steps. Let point I be in cluster L1. If L1 is in the live set, do step 4a; otherwise, do step 4b.

Step 4a. Compute the minimum of the quantity, $R2 = [NC(L) * D(I, L)^2]/[NC(L) + 1]$, over all clusters $L(L \neq L1,\ L = 1,2, \dots, K)$ where the number of points in cluster L is denoted by $NC(L)$; while the number of points in cluster $L1$ is $NC(L1)$; $D(I, L)$ is the Euclidean distance between point I and cluster L; $D[I, L(I)]$is the Euclidean distance between I and the cluster mean of the cluster containing I; $D(I, L)^2$ is the squared Euclidean distance between point I

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

and cluster L. Let L2 be the cluster with the smallest R2. If this value is greater than or equal to $R1 = [NC(L1) * D(I, L1)^2]/[NC(L1) - 1]$, no reallocation is necessary and L2 is the new $IC2(I)$. Otherwise, point I is allocated to cluster L2, and L1 is the new IC2 (I). Cluster centres are updated to be the means of points assigned to them if reallocation has taken place. The two clusters that are involved in the transfer of point I at this particular step are now in the live set.

Step 4b. This step is the same as step 4a, except that the minimum R2 is computed only over clusters in the live set.

Step 5. Stop if the live set is empty; otherwise, go to step 6; after one pass through the data set.

Step 6. This is the quick-transfer (QTRAN) Stage: Consider each point $I(I = 1, 2, \dots, M)$ in turn. Let $L1 = IC1(I)$ and $L2 = IC2(I)$. It is not necessary to check point I if both the clusters $L1$ and $L2$ have not changed in the last M steps. Compute the values:

$$R1 = \frac{[NC(L1) * D(I, L1)^2]}{[NC(L1) - 1]} \text{ and } R2 = [NC(L2) * D(I, L2)^2]/[NC(L2) + 1]$$

If R1 is less than R2; point I remains in cluster $L1$. Otherwise, switch $IC1(I)$ and $IC2(I)$ and update the centres of clusters $L1$ and $L2$. The two clusters are also noted for their involvement in a transfer at this step.

Step 7. If no transfer took place in the last M steps, go to step 4, otherwise, go to step 6.

Algorithm 4: The Hartigan and Wong's (Conventional) Algorithm.

1. Choose the number of clusters, k, and tentative centroids $c_1, c_2, \dots, c_k$.
2. Observe an entity $i \in I$ coming either randomly or according to a pre-specified (dynamically) changing order.
3. $d_{ij} =$ distance between case i and cluster j;
4. $d_{ij} = arg\ arg\ min\ _{1 \le j \le k} d_{ij}$
5. Assign cases *i* to cluster n_i ;
6. Re-compute the cluster means of any changed cluster above;
7. If no further change of cluster membership occurs in a complete iteration; go to step 8,
8. Output results.

Likas' Method

Likas et al. (2003**)** proposed a global k-means clustering algorithm, which constitutes a deterministic effective global clustering algorithm for the minimization of the clustering error that employs the basic k-means algorithm as a local search procedure.

The algorithm proceeds in an incremental way, which helps in solving a clustering problem with k clusters; all problems that are intermediate with $1,2, \dots, k-1$ clusters are sequentially solved. The basic idea behind the global k-means algorithm is that an optimal solution for a clustering problem with k clusters can be obtained by carrying out a series of local searches using the Basic k-means algorithm. At each local search, the $k-1$ cluster centres are always initially placed at their optimal positions corresponding to the clustering problem with $k-1$ clustering (Gan et al., 2007). The remaining kth cluster centre is initially placed at several positions within the data space. Since for $k=1$ the optimal solution is known, it can be iteratively applied to the above procedure to find optimal solutions for all m-clustering problems $m=1,\ldots,K$. (Likas et al., 2003). The global k-means algorithm is described as follows: Suppose that $X=\{x_1,x_2,\ldots,x_n\}$, $x_n\in R^d$ be a given data set in a d-dimensional space. The k-clustering problem aims at partitioning the dataset into k disjoint subsets (clusters) $c_1,c_2,\ldots,c_k$, such that the clustering criterion is optimized. The most widely used clustering criterion is the sum of the squared Euclidean distances between each data point, x_i, and the centroid, m_j. This criterion is called clustering error and it depends on the cluster centre, $m_1,m_2,\ldots,m_k$:

$$E(m_1,m_2,\ldots,m_k)=\sum_i^n \quad \sum_j^k \quad d_{euc}^2\left(x_i,m_j\right) \qquad (4)$$

In Equation (4) above, x_i and m_j, are the data point and the cluster centre (centroid) while $d_{euc}^2(.,.)$ is the squared Euclidean distance which is one of the most widely used clustering criteria. This method does not depend on any initial values. Instead of selecting initial values randomly for all cluster centres as is the case with most global clustering algorithms, the method proceeds in an incremental way attempting to optimally add one new cluster centre at each stage of the iteration.

To be more specific in solving a clustering problem with k clusters, the iterative method is as follows; we start with $k=1$ cluster and find its optimal position that corresponds to the centroid of the data set, x. To solve the problem with two clusters $(k=2)$, we perform N executions of the k-means algorithm from the following initial positions of the cluster centres: the first cluster centre is always placed at the optimal position for the problem with $k=1$, while the second centre at execution is placed at the position of the data point, x_n $(n=1,2,\ldots,N)$. The best solution obtained after the N executions of the Basic k-means algorithm is considered as the solution for the clustering problem with $k=2$. In general, let $[m_1^*(k),m_2^*(k),\ldots,m_K^*(k)]$denote the final solution for the k-clustering problem. Immediately the solution of the k-clustering problem has been found, we try to find the solution of the K-clustering problem as follows: we perform N runs of the k-mean algorithm with k clusters where each run n starts from the initial state$[m_1^*(k-1),m_2^*(k-1),\ldots,m_{K-1}^*(k-1),x_n]$.

The best solution obtained from the N runs is considered as the solution $[m_1^*(k),m_2^*(k),\ldots,m_K^*(k)]$ of the K-clustering problem. By proceeding in the above fashion, we finally obtain a solution with m clusters having also found a solution for all K-clustering problems with $k<m$ (Likas et al., 2003). The global K-means algorithm for the computation of $q\leq n$ cluster in the data set A can be described as follows.

Algorithm 2.5: The Likas (Global) Algorithm.

1. (Initialization) compute the centroid x_1, of the set A:

$$x_1 = \frac{1}{n}\sum_{i=1}^{n} \quad m_i,\ m_i \in A, \qquad i = 1,2,\cdots,n \; and \; set \; k = 1.$$

2. Set $k = k + 1$ and consider the centres,$x_1, x_2, \cdots, x_{k-1}$, from the previous iteration.
3. Consider each point m, of A as a starting point for the k cluster centre; thus obtain n initial solution with k points ($x_1, x_2, \cdots, x_{k-1}$, m); apply Basic K-means algorithm to each of them; keep the best k-partition obtained and its centres, $x_1, x_2, \cdots, x_k$.
4. If $k = n$ then stop, otherwise go to step 2.

Faber's Method

Faber (1994) proposed Faber's method which is popularly known as the continuous k-means algorithm. The continuous k-means algorithm is faster than the standard k-means algorithm and it is also different from the standard k-means algorithm in two ways. First, the reference points in the continuous k-means algorithm are chosen as a random sample from the whole population of data point, while in the standard k-means algorithm the initial reference points are chosen more or less arbitrarily. Secondly, the way the data points are treated during the update process. During the iteration, the standard k-means algorithm examines all of the data points in sequence while the continuous k-means algorithm examines only a random sample of data points. If the data set is very large and the sample is representative of the data set, the continuous k-means algorithm should converge much faster than the algorithm that examines every point in the sequence. To be precise, the continuous k-means algorithm adopts MacQueen's method of updating the centroids during the initial partitioning, when the data points are first assigned to clusters (Faber, 1994).

Theoretically, random sampling represents a return to Macqueen's original concept of the algorithm as a method of clustering data over continuous space. In Macqueen's formulation, the error measure E_i for each region R_i is given by

$$E_i = \int_{x \in R_i} \quad f(x) \parallel x - z_i \parallel^2 dx \tag{5}$$

where $f(x)$ is the probability distribution function, which is a continuous function defined over the space, x is the data point and z_i is the centroid of the region R_i; while E_i is the total error measure. Hence, a large set of discrete data point can be seen as a large sample as well as a good estimate of the continuous probability density $f(x)$. Then it suffices that a random sample of the data set can also be a good estimate of $f(x)$. Such a sample yields a representative set of cluster centroids and a reasonable estimate of the error measure without using all the points in the original data set.

New K-means Method

This method uses the Minkowski's distance, or r-metric, between vectors or N-dimensional points where $y = (y_v)$ and $c = (c_v)$ which is defined by the formula

$$d(y,c) = [\sum_{v=1}^{N} \quad |y_v - c_v|^r]^{\frac{1}{r}} \tag{6}$$

In Equation (6), y_v are data points, c_v are cluster centres (centroids) and $\sum_{v=1}^{N} \quad |x_v - y_v|^r$ is the r Minkowski distance. In application, when values $r = 2$ (Euclidean metric), $r = 1$ (Manhattan, or city block, metric) and $r \to \infty$ (Chebyshev, or Maximum, metric). However, the Euclidean k-means criterion is the usual k-means when $r = 2$ which is stated as

$$E = W(s,c) = \sum_{k=1}^{K} \quad \sum_{i=s_K} \quad d_{euc}^2(y_i, c_k)$$

where k represents the number of clusters, $c_k \epsilon\ c = \{c_1, c_2, \dots, c_k\}$ is the centroid of cluster s_k, $d_{euc}^2(y_i, c_k)$ is the squared Euclidean distance between an entity (cluster point) $y_i \epsilon\ s_k$ and its respective centroid c_k. The Minkowski k-means criterion allows the use of any distance function and W(s,c) is the square error criterion which is the sum of values over all clusters. Focusing on the Minkowski metric, which is between the N-dimensional entities y_i and c_k and is defined by

$$d(y_i, c_k) = [\sum_{V=1}^{N} \quad |y_{iv} - c_{kv}|^r]^{\frac{1}{r}} \tag{7}$$

r is the exponent or power of Equation (7) which becomes

$$W_r(s,c) = \sum_{k=1}^{K} \quad \sum_{i=s_K} \quad d^r(y_r, c_k) = \sum_{k=1}^{K} \quad \sum_{i\epsilon S_k} \quad \sum_{v=1}^{N} \quad |y_{iv} - c_{kv}|^r \tag{8}$$

This method is a batch k-means algorithm in which the minimum distance rule applies with the distance being the r power of Minkowski r-metric rather than the squared Euclidean distance (Amorim and Komisarczuk, 2012; Amorim, 2012; Amorim and Mirkin, 2012).

Algorithm 6: The New K-Means Algorithm.

1. Choose at random the number of cluster centres (centroids) $c = c_1, c_2, \dots, c_k$.

2. Calculate the distance between each data point and cluster centres using Equation (7)

3. Assign data point to the cluster centre whose distance from the cluster centre is the minimum of all cluster centres.

4. New cluster centre is calculated using $v_i = \frac{1}{|C_i|} \sum_{y \epsilon c_i} \quad y_i$ where $|c_i|$ denotes the absolute value of data points in ith cluster and v_i is the mean of the cluster c_i and $\sum \quad y_i$ is the sum of points or cases in the data space.

5. The distance between each data point and new obtained cluster centres is recalculated.

6. If no data point was reassigned then stop, otherwise repeat step 3 to 5.

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

RESULTS AND DISCUSSION

This section shows the performance comparison of the modified k-means method and the existing six k-means clustering methods using R statistical software (R version 3.2.2) support window 64-bit system. We conducted experiments using one simulated data set and two real-life data sets to ensure the efficiency of the proposed k-means method. The numbers of clusters k used are two and three since research has proven that the optimal number of clusters k will either be two, three, or four using methods like elbow, the silhouette and the gap statistic methods (Kaufman and Rousseeuw, 1990).

The performance of the proposed method was evaluated using total intra-cluster variance and accuracy parameters, after which was ranked.

Total intra-cluster variance: The total intra-cluster variance is defined as the sum of squared distance between points and the corresponding centroid. That is; $W(C_K) = \sum_{x_i \epsilon c_k} \quad (x_i - \mu_k)^2$ where

- x_i is the data point belonging to the cluster c_k.
- μ_k is the mean value of the points assigned to the cluster c_k.

Accuracy: Accuracy is defined as the ratio of the total number of correctly classified instances divided by a total number of correctly plus incorrectly classified instances denoted by Acc. (%).

Simulated Data

The simulated data was generated randomly from a Gaussian (Normal) distribution with a dimension of 300 rows and 2 columns (categories or attributes) that are divided into two and three clusters (that is, k = 2, 3). We chose 300 true centres uniformly at random given the above dimension. The point from the Gaussian distributions has a variance of 1 around each true centre. Thus, this obtained several well-separated Gaussians with the true centres providing a good approximation to the optimal clustering.

Shown below is the summary table of the results of experiments and data analysis of six existing methods when the number of clusters k is two and three respectively:

Table 1: Summary results of simulated data when the number of clusters k = 2 and 3.

Methods	When K = 2				When K = 3				Combined Rank
	μ	σ	Acc. (%)	Rank	μ	σ	Acc. (%)	Rank	
Forgy	1.58	0.49	80.8	4	2.25	0.78	81.4	4	8
Lloyd	1.50	0.50	79.1	6	1.92	0.81	79.0	5	11
MacQueen	1.50	0.50	79.1	6	2.30	0.75	83.7	3	9
Hartigan & Wong	1.50	0.50	79.1	6	2.14	0.83	78.3	6	12
Likas	1.78	0.39	89.0	1	2.54	0.68	88.6	1	2
Faber	1.76	0.43	83.3	3	2.05	0.92	72.0	7	10
Proposed Method	1.51	0.42	86.8	2	1.54	0.69	87.5	2	4

From the above results of the simulation generated randomly, when the number of clusters k = 2, the Likas' method performed best with a minimum standard deviation of 0.39 and a high accuracy rate of 89 per cent, followed by the proposed method with a minimum standard deviation of 0.42 and accuracy rate of 86.8 because the variance (the total within-cluster sum of squares) is minimized; it measures the compactness (i.e. goodness) of the clustering which is meant to be as small as possible, also, high accuracy indicates how better the method is. When the number of clusters k = 3, Likas' method also performed best with a standard deviation of 0.68 and accuracy rate of 88.6 per cent, followed by the proposed method with a standard deviation of 0.690 and an accuracy rate of 87.5 per cent which performed better than the other five existing methods. From the combined ranking, the proposed method was second-best in performance.

Real-Life Data

To understand how efficient these methods are under more practical circumstances, we run several experiments on two data sets which consist of the iris data set, and the yeast cell cycle data set. The two data sets are from UC-Irvine Machine Learning Repository namely: the iris data set and the wine data set. Each experiment involves solving the k-means problem on a set of points in a real dimensional space.

Iris Data Set

The iris flower data set is a multivariate data set with 150 rows (instances) which are divided into 3 instances each, where each class refers to a type of iris plant (iris setosa, iris versicolor, and iris virginica): the number of columns (attributes) is 4 which consist of sepal length, sepal width, petal length and petal width (Fisher, 1936). The summary table of the results of the experiments when the numbers of clusters k- two and three are shown in Table 2 below:

Table 2: Summary results of iris data when the number of clusters k = 2 and 3.

Methods	When K = 2				When K = 3				Combined Rank
	μ	σ	Acc. (%)	Rank	μ	σ	Acc. (%)	Rank	
Forgy	1.35	0.48	83.5	5	1.56	0.81	82.0	5	10
Lloyd	1.65	0.48	83.5	5	2.49	0.74	85.2	3	8
MacQueen	1.35	0.48	83.5	5	1.93	0.60	91.5	1	6
Hartigan & Wong	1.65	0.48	83.5	5	2.08	0.86	79.1	6	11
Likas	1.63	0.48	84.7	5	2.65	0.72	86.0	2	7
Faber	1.86	0.35	89.2	2	2.47	0.92	77.4	7	9
Proposed Method	1.78	0.33	89.7	1	1.95	0.80	82.4	4	5

From the above experiments and summary table on the iris data set, it is observed that when the number of clusters k = 2, the proposed method performed better than the other existing methods with a standard deviation of 0.33 and an accuracy of 89.7 per cent. Also, when the

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

number of clusters k = 3, MacQueen's method performed better than every other method with a standard deviation of 0.60 and 91.5 per cent accuracy; while the proposed method performed better than Faber's method, Forgy's method and Hartigan & Wong's method with a minimum standard deviation of 0.80 and 82.4 per cent accuracy. Using the combined ranking, the proposed method was observed the best in the iris data set.

Wine Data Set

The wine data set is multivariate data with 178 rows (instances) and three classes with 13 attributes (columns). The attributes of the data set are alcohol, malic acid, ash, alkalinity of ash, magnesium, phenols, flavonoids, non-flavonoid phenols, proanthocyanins, colour intensity, hue, 0D280/0D315 of diluted wines and proline. The output of the experiments when the number of clusters k = 2 and 3 will be summarized in Table 3 below:

Table 3: Summary results of wine data when the number of clusters k = 2 and 3.

Methods	When K = 2				When K = 3				Combined Rank
	μ	σ	Acc. (%)	Rank	μ	σ	Acc. (%)	Rank	
Forgy	1.08	0.269	91.4	5	2.59	0.78	67.8	7	12
Lloyd	1.92	0.268	91.8	3	2.17	0.49	86.3	2	5
MacQueen	1.08	0.268	91.8	3	2.13	0.43	88.1	1	4
Hartigan & Wong	1.92	0.268	91.8	3	1.24	0.50	83.8	3	6
Likas	1.33	0.339	86.7	7	2.73	0.64	74.8	5	12
Faber	1.86	0.346	87.2	6	1.87	0.66	73.5	6	12
Proposed Method	1.91	0.266	92.1	1	2.44	0.63	76.2	4	5

It was observed that when the number of clusters k = 2, the proposed method performed better than the other methods with a minimal standard deviation of 0.266 and an accuracy of 92.1 per cent. When the number of clusters k = 3, MacQueen's method outperformed every other method with a standard deviation of 0.4310 and an accuracy of 88.10 per cent. The performance of the proposed method was relatively efficient than Forgy's method, Likas' method and also that of Faber's method with a standard deviation of 0.63 and accuracy of 76.2 per cent. From the combined ranking of the wine data set, our proposed method is the second-best in minimizing the intra-cluster variance.

CONCLUSION

In this paper, we have presented a new k-means clustering method that uses Minkowski's distance in calculating between each data points and the cluster centroids which performed favourably in comparison with existing methods in terms of minimizing the total intra-cluster variance. From the experimental summary results considering the combined ranking, the new k-means method was effective than most existing methods both in simulation and real-life data sets used when the number of clusters k = 2 and 3.

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

Acknowledgements

The authors wish to thank the referees for their worthwhile comments and suggestions.

REFERENCES

Amorim, R. C. 2012. Constrained clustering with Minkowski weighted k-means. Proceedings of the 13th IEEE International Symposium on Computational Intelligence and Informatics, 13-17.

Amorim, R. C., Komisarczuk, P. 2012. On Initializations for the Minkowski weighted k-means. International Symposium on Intelligent Data Analysis, 45-55.

Amorim, R. C., Mirkin, B. 2012. Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. Pattern Recognition, 45 (3), 1061-1075.

Anderberg, M. R. 1973. Cluster Analysis for Applications. New York: Academic Press.

Everitt, B., Landau, S., Leese, M., Stajl, D. 2011. Cluster Analysis, 5thedition, John Wiley and Sons.

Faber, V. 1994. Clustering and the continuous k-means algorithm: Los Alamos Science, 22, 138-144.

Fisher, R. A. 1936. ”The Use of Multiple Measurements in Taxonomic Problems, “Annals of Eugenics, 3, 179-188.

Forgy, E. W. 1965. Cluster analysis of multivariate data: efficiency versus interpretability of classification. Biometrics, 21, 768-769.

Gan, G., Ma, C., Wu, J. 2007. Data Clustering: Theory, Algorithms, and Applications, SIAM Series.

Gunopulos, D., Das, G. 2001. Time series similarity measures and time series indexing. Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 624. New York: ACM Press.

Hartigan, J. A. 1975. Clustering Algorithms. New York: John Wiley and Sons.

Hartigan, J. A., Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics), 28 (1), 100-108.

Johnson, R. A., Wichern, D. W. 2002. Applied Multivariate Statistical Analysis: 5th Edition, Eaglewood Cliffs, NJ: Prentice-Hall.

Kaufman, L., Rousseeuw, P. J. 1990. Finding Groups in Data, An Introduction to Cluster Analysis. Wiley Series, New York: John Wiley and Sons.

Likas, A., Vlassis, N., Verbeek, J. 2003. The global k-means clustering algorithm. Pattern Recognition, 36 (2), 451-461.

Lloyd, S. 1982. Least squares quantization in PCM. IEEE Transaction on Information Theory, 28 (2), 129-137.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations, In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, (1), 281-297. Berkeley, CA: University of California Press.

Morissette, L., Chartier, S. 2013. The k-means clustering technique: General considerations and implementation in Mathematica. Tutorials in Quantitative Methods for Psychology, 9 (1), 15-24.

Article DOI: 10.52589/BJCNIT-XEPSJBWX
DOI URL: https://doi.org/10.52589/BJCNIT-XEPSJBWX

Peńa, J., Lozano, J., Larrańaga, P. 1999. An empirical comparison of four initialization methods for the k-means algorithm: Pattern Recognition Letters, 20 (10), 1027-1040.

Slonim, N., Aharoni, E. and Crammer, K. (2013). Hartigan's K-Means Versus Lloyd's K-Means-Is It Time for a Change? Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 1677-1684.