2023-08-01

Robust Mahalanobis K-Means Algorithm in Comparison with Other Existing Clustering Methods.

Eleazer Tabi Serebour
University of Texas at El Paso

ROBUST MAHALANOBIS K-MEANS ALGORITHM IN COMPARISON WITH OTHER EXISTING CLUSTERING METHODS

ELEAZER TABI SEREBOUR

Master's Program in Mathematical Sciences

APPROVED:

Abhijit Mandal, Ph.D, Chair.

Suneel Babu Chatla, Ph.D.

Mohammad Iqbal H. Bhuiyan, Ph.D.

Stephen Crites, Ph.D
Dean of the Graduate School

to my

MOTHER, Mrs. Agarthar Dansowaa

with much love

ROBUST MAHALANOBIS K-MEANS ALGORITHM IN COMPARISON WITH OTHER EXISTING CLUSTERING METHODS

by

ELEAZER TABI SEREBOUR

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Mathematical Sciences

THE UNIVERSITY OF TEXAS AT EL PASO

August 2023

Acknowledgement

I am immensely grateful to the Almighty Lord for making this journey a success. I also want to express my sincere gratitude to my thesis advisor, Dr. Abhijit Mandal from The University of Texas at El Paso's Department of Mathematical Sciences, for his irreplaceable advice and assistance throughout the duration of this thesis. Moreover, I want to express my sincere thanks to Dr. Suneel Baabu Chatla from the Department of Mathematical Sciences and Dr. Mohammad Iqbal H. Bhuiyan from the School of Pharmacy at The University of Texas at El Paso for their dedicated efforts and support.

Abstract

This study enhances K-means Mahalanobis clustering using Density Power Divergence (DPD) for outlier handling and detection. Through the utilization of simulations and the analysis of real-world data, our approach consistently outperforms standard K-means, Mahalanobis K-means, Fuzzy C-means, and others in clustering datasets with outliers. While our method performs similarly to others on spherical datasets, it ranks second to DBSCAN for arbitrary shapes. We showcase its superiority on real-life datasets (Iris flower and wheat seed), demonstrating resilient outlier identification. By navigating various structures and cluster characteristics, our Modified Mahalanobis K-means method proves adaptable and robust, offering insights into diverse clustering scenarios. The study explores robust clustering to mitigate outlier impact in statistical analysis, contributing to improved clustering in outlier-prone datasets.

Keywords: Clustering, Performance Metrics, outliers, Density Power Divergence, Mahalanobis DPD

Contents

Page

List of Tables

List of Figures

Chapter 1

Introduction

Cluster analysis is a statistical technique in data analysis that is used to identify groups with similarities in a dataset. Cluster analysis aims to group observations such as respondents, products, or other objects based on their shared characteristics. The goal of cluster analysis is to carve up the data into groups that are homogeneous within each group but heterogeneous across groups. Thus, observations within each group are similar to each other and different from observations in other groups (Sharma, 1995). Cluster analysis does not assume any specific number of groups or group structure but rather observations are grouped based on their similarities or differences. Dataset grouping can help to identify patterns,outliers and potential relationships between them.

Outliers are data points that deviate significantly from the remaining observations in a given dataset. These data points can have a major impact on data analysis, as they can skew the results and lead to incorrect conclusions. Outliers can arise due to several reasons, including data entry errors, measurement errors or genuine data inconsistencies. Identifying and handling outliers in a dataset is an important task in data analysis, as it can significantly affect the accuracy and dependability of the finding. Descriptive statistics such as the mean and variance can be affected by outliers. Outliers can also affect the performance of machine learning algorithms, such as classification, regression, and clustering. The presence of outliers in a dataset can significantly affect clustering results leading to incorrect inferences. Traditional clustering methods are not generally resistant to outliers and can be completely affected by outliers (Aggarwal et al., 2015). An illustration of this phenomenon is when different groups may appear as a single cluster or when spuriously-formed clusters contain only outlier observations. These difficulties also emerge when utilizing data mining methods that incorporate clustering procedures in “unsupervised learning” as large, high-dimensional, and automated dataset tend to comprise outliers. Robust clustering is a formindable technique that can reduce the effects of outliers and churn-out more reliable and accurate clustering outcomes.

Robust clustering technique is designed to minimize the impact of outliers on statistical analysis. Robust clustering is notably valuable in cases where conventional statistical methods are violated, especially when the data involves non-normal distribution or has outliers. Robust clustering can also facilitate the identification of dataset that are more resilient to outliers and enable the estimation of clustering algorithm parameters in a manner that is less susceptible to the influence of outlier observations (García-Escudero et al., 2008). In this thesis, we will explore the use of robust clustering methods on dataset with outliers. We will investigate the impact of outliers on the clustering results and compare the performance of traditional clustering algorithms with robust clustering algorithms in the presence of outliers.

1.0.1 Application of Cluster Analysis

Clustering is a powerful technique used in various fields to group data points with similar characteristics. For instance, in the education sector, clustering can be used to identify groups of students with similar academic performance or behavioral patterns. In the financial industry, clustering can be applied to group customers with similar investment behaviors or risk appetites. In agriculture, clustering can be used to group farms based on their soil type, weather patterns, and crop yields. In genetics, clustering can be used to group individuals with similar genetic profiles to aid in disease diagnosis and treatment. Cluster analysis can be used to identify communities or groups within social networks, such as Facebook, Instagram or Twitter. This can help researchers to better understand the structure and dynamics of social networks and how information spreads through them. Overall, clustering is a versatile technique that can be used to identify patterns and relationships within data in a variety of fields.

1.0.2 Types of Clustering

There are numerous clustering algorithms, which can be grouped into seven general categories. The first category is partition-based clustering, which splits the dataset into a fixed number of clusters. The k-means algorithm is the most widely used partition-based algorithm. The second type is hierarchical clustering, which creates a hierarchy of nested clusters by merging or splitting clusters recursively. Examples include Ward's method and Single Linkage clustering. The third type is density-based clustering, which clusters data points based on the density of nearby points. The most well-known density-based algorithm is the DBSCAN algorithm. The fourth type is distribution-based clustering, which assumes that the data is generated from a probability distribution and models the clusters based on statistical properties such as mean and covariance. The fifth type is grid-based clustering, which partitions the data space into a set of cells and groups points based on their location in the cells. The sixth type is model-based clustering, which models the underlying distribution of the data using a probabilistic model and assigns data points to clusters based on their likelihood of belonging to a cluster. Finally, the seventh type is constraint-based clustering, which involves integrating user-specified constraints into the clustering process to guide the formation of clusters. Each type of clustering algorithm has its own advantages and disadvantages, and the decision of which algorithm to use depends on the nature of the data and the specific problem being addressed (Tan et al., 2006).

1.0.3 Distance Measures

There are several distance measures commonly used to determine the similarity or dissimilarity between data points. The choice of distance measure depends on the nature of the data and the specific requirements of the research. Here are a few commonly used distance measures for clustering:

- Minkowski Distance

It is the generalization of both Euclidean and Manhattan distances. It is defined as:

$$D = \left(\sum_{i=1}^{n} |x - u|^{n} \right)^{1/n}. \tag{1.1}$$

- Euclidean Distance

  This is the most widely used distance measure in clustering. It calculates the straight-line between two points in Euclidean space. Expressed as:

$$D_e(x, u) = \|x - u\| = \sqrt{\sum_{i=1}^{n} (x_i - u_i)^2}. \tag{1.2}$$

- City-block Distance

  It is a special case of Minkowski metric at $n = 1$. It tends to form hyperrectangular clusters. It is suitable when dealing with data that has a grid-like structures. Mathematically expressed as:

$$D_c = \sum_{i=1}^{n} |x_i - u_i|. \tag{1.3}$$

- Mahalanobis Distance

  It is defined as:

$$D_m(x, u) = \sqrt{\sum_{i=1}^{n} (x_i - u_i)^T \, \Sigma^{-1} \, (x_i - u_i)}. \tag{1.4}$$

  where Σ is the covariance matrix. Mahalanobis distance can be used as a similarity measure to group points that close to each other and are related. It accounts for correlation. When dataset are not correlated, squared Mahalanobis distance is equivalent to squared Euclidean distance. Mahanalobis sometimes cause computational burden.

Chapter 2

Literature Review

The importance of robust clustering methods in practical statistical applications was highlighted in (García-Escudero et al., 2010), which aim to avoid unsatisfactory results caused by deviations from theoretical assumptions and the presence of outlying observations. The review focuses on approaches based on trimming, which discard outlying data during the clustering process. Shah and Koltun (2017) described a new clustering algorithm that achieves high accuracy across multiple domains, scales efficiently to high dimensions and large datasets, and can be integrated into end-to-end feature learning pipelines. The method is evaluated on various large datasets and outperforms the best prior algorithm by a factor of 3 in average rank. The paper (Davé and Krishnapuram, 1997) discussed the analysis and unification of several popular robust clustering methods and establishes connections between fuzzy set theory and robust statistics, ultimately proposing generic algorithms and guidelines for clustering noisy data. Banerjee and Davé (2012) identified robust model fitting in methodological and algorithmic advances, and includes examples of robust clustering applications to synthetic and real-world datasets. Notsu and Eguchi (2016) proposed a robust clustering method called Gamma-clust that uses a robust estimation for cluster centers based on gamma-divergence to handle contamination of scattered observations, providing superior results compared to other robust clustering methods in simulations and data analysis. (Anum and Pokojovy, 2023) introduced a novel optimization method combining Newton's method and gradient descent to solve a challenging mathematical problem involving Gaussian parameters. They prove its global convergence and efficiency in comparison to the widely-used Minimum Covariance Determinant estimator. Application to real health data demonstrates its practicality. Yang et al. (2012) proposed a robust EM clustering algorithm for Gaussian mixture models, which addresses the sensitivity of the EM algorithm to initialization and the need for a priori specification of the number of components. The proposed method automatically obtains an optimal number of clusters and is shown to outperform existing clustering methods in experimental examples. The paper (Li et al., 2022) proposed a robust clustering method based on directed k-nearest neighbor graph, called CDKNN, which automatically identifies the desired cluster number and produces reliable clustering results on nonlinear and locally tight-connected data patterns, and it outperforms other clustering methods in terms of clustering accuracy.

Clustering has gained more significance in the era of big data in recent years. As datasets become larger and more complex, traditional statistical techniques may not be adequate in identifying patterns and structures, which makes clustering a desirable substitute. Clustering algorithms can effectively manage big datasets, detect outliers, and identify nonlinear relationships among variables. In this section, we will consider some existing clustering methods.

2.1 Hierarchical Clustering

5r4r5Hierarchical clustering is an unsupervised learning algorithm used to group similar data points or objects into clusters based on their similarity or dissimilarity. It can be grouped into two types: agglomerative and divisive. With Agglomerative clustering, it begins with each data point as a separate cluster and then merges the most similar clusters into larger ones till all data points are in a single cluster. Divisive clustering starts with all data points in a single cluster and then splits the cluster into two recursively until each data point is in its own cluster. The crucial point is how best to choose the next cluster(s) to split or merge. There are many applications in various fields, including biology, social sciences, and computer science when it comes to Hierarchical clustering. It is used for tasks such as gene expression analysis, customer segmentation, and image clustering. Agglomerative clustering is more preferably used than divisive clustering due to its computational efficiency, ease of implementation, and ability to generate a dendrogram. However, divisive clustering can be useful in certain scenarios, such as when the dataset contains a large number of features or when the clusters have a well-defined shape.

2.1.1 Agglomerative Clustering

Agglomerative clustering has several types based on the criteria used to decide which clusters to merge at each step. This includes, single-linkage, complete-linkage, average-linkage, and Ward's method. Single-linkage clustering defines the distance between two clusters as the distance between their closest data points and tends to form long, skinny clusters that can be sensitive to noise and outliers. Complete-linkage clustering defines the distance between two clusters as the distance between their farthest data points and tends to form compact, spherical clusters that are less sensitive to noise and outliers compared to single-linkage clustering. Average-linkage clustering defines the distance between two clusters as the average distance between all their data point pairs and strikes a balance between the sensitivity to noise and the formation of compact clusters. Ward's method defines the distance between two clusters as the increase in the sum of squared distances within each cluster after the merge and tends to form clusters that minimize the sum of squared distances within each cluster, resulting in more balanced clusters. Divisive clustering is preferred when the clusters have highly irregular shapes or sizes, and the goal is to understand the detailed structure of the data. Hierarchical clustering is widely used in various fields for tasks such as gene expression analysis, customer segmentation, and image clustering (Johnson et al., 2002).

Agglomerative Hierarchical Clustering Algorithm

The agglomerative hierarchical clustering algorithm consists of several steps for grouping N objects (items or variables).

- Begin with N clusters, each containing a single item, and a distance matrix $D = d_{ik}$

that is $N \times N$ and symmetric, representing the distances or similarities between objects.

- The nearest (most similar) pair of clusters in the distance matrix should be searched. Let d_{uv} be the distance between these clusters, U and V.
- Integrate clusters U and V to form a new cluster labeled UV. Revise the distance matrix by removing the rows and columns corresponding to U and V and adding a row and column representing the distances between UV and the remaining clusters.
- Steps 2 and 3 repeats $N-1$ times. After this, all objects will be a single cluster. Keep track of the merged clusters and the distances or similarities at which they merged (Sharma, 1995).

2.1.2 Advantages of Agglomerative hierarchical Clustering

Agglomerative clustering is a flexible and scalable method that can handle different types of data, including large datasets, generating a visual hierarchy of clusters for easy identification of subgroups. It does not require the user to specify the number of clusters beforehand and preserves the original data points.

2.1.3 Disadvantages of Agglomerative Hierarchical Clustering

However, Agglomerative clustering has some disadvantages, including high computational complexity for large datasets, sensitivity to noise and outliers, bias with missing data, and struggles with high-dimensional data (Tufféry, 2011).

2.2 K-means Clustering

K-means is among the clustering method that can be widely used to solve various clustering problems across different fields and applications. The K-means clustering divides data observations into k clusters based on their similarities, where each data point is assigned to the cluster group with the highest similarity. K-means comes along with a simple approach where a dataset is classified into a fixed number of clusters, k, using a predetermined value. The primary measure of distance for K-means clustering is the euclidean distance. This method calculates the distance between two observation in n-dimensional space as the square root of the sum of the squared differences between their corresponding coordinates. The Euclidean distance is mostly preferred in K-means clustering because of its ability to simplify the computation and work well with continuous. However,depending on the characteristics of the data and the goals of the analysis, other measures of distance, such as Manhattan and Mahalanobis can be used. The main goal of K-means clustering is to reduce the overall sum of squared distances between each data point and the centroid it belongs to. In other words, it seeks to group dataset into k clusters such that the sum of squared distances between each data point and its assigned centroid is minimized. Mathematically, the objective function of the K-means algorithm can defined as:

$$\text{SSE} = \sum_{i=1}^{k} \sum_{x \in C_i}^{n} \left\| x - \mu_i \right\|^2 . \tag{2.1}$$

The K-means clustering strives to minimize the sum of squared errors (SSE) for all clusters. Here, k denotes the cluster index, x represents the i^{th} data point in the k^{th} cluster, c_i is the set of data points in the k^{th} cluster, u_i is the centroid of the k^{th} cluster, and $\|x - \mu_i\|^2$ denotes the squared Euclidean distance between a data point and its corresponding centroid (Yuan and Yang, 2019).

Determining the parameter of K-means Clustering

The two commonly used techniques to determine the optimal number of clusters in K-means clustering are the elbow method and silhouette analysis. The elbow method involves plotting WCSS against k and selecting the value of k where the decrease in WCSS starts to level off, indicating further subdivision into additional clusters would not greatly improve clustering performance. Silhouette analysis calculates a measure of similarity and dissimilarity for each data point and selects the optimal number of clusters that maximizes the average silhouette width. Multiple runs of K-means clustering with different initial centroid placements can be performed to address the impact of initial centroid placement on clustering performance (Kodinariya et al., 2013).

K-means Algorithm

The K-means clustering approach is a straightforward method, and we will start by explaining the algorithm. Firstly, we select K initial centroids, where k is a pre-defined parameter representing the number of desired clusters. Next, each data point is assigned to the nearest centroid, and these groups of data points assigned to a centroid are clusters. The centroid of each cluster is updated based on the data points assigned to it. The assignment and updating of the centroids are repeated until no data point changes its assigned cluster or until the centroids remain the same. The K-means algorithm can be executed in the following steps:

- Choose k clusters and randomly set their centroids
- Assign each data point to the closest centroid based on the Euclidean distance.
- Recalculate the centroids by taking the average of all the data points assigned to each centroid.
- Iteratively perform steps 2 and 3 until the algorithm converges or reaches a maximum number of iterations.
- Ultimately, the K-means algorithm produces the k clusters along with their respective centroids.

2.2.1 Advantages of K-means Clustering

K-means clustering is a versatile algorithm that is easy to use and efficient, making it suitable for a wide range of data types. It can handle numeric data effectively and has multiple runs to ensure accurate results. Variations such as bisecting K-means are even more effective, with fewer initialization problems.

2.2.2 Disadvantages of K-means Clustering

K-means clustering, however, has some limitations. Incorporating categorical variables is challenging, which limits its use in datasets with mixed variable types. Where large outliers are in existence, the algorithm becomes less effective since it is sensitive to outliers. High-dimensional datasets can also cause issues, requiring feature selection or dimensionality reduction algorithms. Although pure subclusters can be pinpoint with enough specified clusters, K-means clustering struggles with non-spherical clusters or clusters with varying densities and sizes, conversely, K-medoid clustering can handle these types of data but is more computationally exhorbitant (Xu and Tian, 2015).

2.3 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN, a clustering algorithm that utilizes density as a basis, is presented by (Ester et al., 1996) in their publication. DBSCAN is a clustering technique that utilizes point density to identify clusters. Regions with high point density are labeled as clusters, while regions with low density stipulates the presence of noise or outliers. DBSCAN is an effective method for analyzing large dataset with noise and can detect clusters of varying shapes and sizes. It distinguishes data points into three categories: core points, border points, and noise points. Core points have a minimum number of other points within a specified radius (ϵ), border points are within the ϵ radius of a core point but do not meet the criteria for a core point, and noise points are those that are not within the ϵ radius of any core point. The DBSCAN's algorithm's main idea is that each point in a cluster's neighborhood must have a minimum density that exceeds a specified threshold. Notably, when utilizing a neighbourhood distance parameter (ϵ) of 1000, the DBSCAN algorithm sucessfully produced clusters during the clustering process (Madhulatha, 2012).

DBSCAN Algorithm

DBSCAN Algorithm shows enhanced homogeneity and variety in personalised clustering tasks when applied to datasets characterized by non-uniform density, broad value ranges, and gradually infrequent patterns. Two parameters is required to be inputed by the user when it comes to DBSCAN algorithm. The first parameter, known as ϵ, defines the radius that outlines the neighborhood area for each point. The second parameter, MinPts, establishes the minimum number of points that need to be present in the ϵ neighborhood.

Determining the optimal parameters in DBSCAN algorithm

On choosing the parameters needed in the DBSCAN algorithm; thus, MinPts and ϵ. MinPts is the least number of data points expected in each cluster, and it should be at least 3 and increased for noisy and large datasets. Choosing the MinPts to twice the number of dimensions of the dataset can be helpful. The optimal ϵ value can be generated using a k-distance plot. K-nearest neighbors are computed based on the value of MinPts considered earlier, and the average distance between each data point and its k-nearest neighbors is calculated. The average distances will then be plotted from lowest to the highest on a graph known. The optimal value of ϵ is determined by identifying the point on the graph where the slope is the greatest. Smaller ϵ values are considered to be best but choosing an excessively low value of ϵ may result in data points that do not belong to any cluster, whereas selecting an overly high value can cause different clusters to be merged (Rahmah and Sitanggang, 2016).

Steps

Assuming that the values of ϵ and MinPts are already given:

- To begin with, select an unclassified point randomly
- The algorithm uses the ϵ value specified by the user to identify directly density-reachable points for each point in the dataset and classify the points as core, border, or noise
- A new cluster is created around it, if the point is a core point, and all directly reachable points within the ϵ radius are added to the cluster. If the point is a border point, it is marked as such:
- The process is repeated for all unclassified core and border points until all points are classified as core, border, or noise
- Noise points are obliterated from the dataset.
- Core points within the ϵ radius are connected and grouped into separate clusters
- Each border point is assigned to the cluster of its associated core points.

2.3.1 Advantages of DBSCAN

DBSCAN is a clustering algorithm that has numerous advantages over other algorithms. It is adaptable to clusters with varying shapes and sizes, and it can detect oddly or irregularly shaped clusters, such as ring-shaped clusters. DBSCAN is also robust to outliers, making it appropriate for datasets with multiple outliers. Furthermore, it can automatically detect the number of clusters in the data and does not require users to specify the number of clusters. DBSCAN is also less sensitive to initialization conditions, and it is relatively fast and optimized for time complexity.

2.3.2 Disadvantages of DBSCAN

In contrast, DBSCAN has some setbacks. Firstly, it is highly sensitive to its input parameters ϵ and MinPts, which can be challenging to select accurately. Secondly, the algorithm can be computationally expensive, especially for large datasets, as it computes the distance between each point and every other point. Additionally, DBSCAN may encounter difficulty with datasets that have clusters of varying densities, potentially resulting in a cluster splitting into multiple clusters or merging with another cluster. Finally, changing the input data may produce different results for the same dataset.

2.4 Fuzzy C-Means (FCM) Clustering

The Fuzzy C-Means (FCM) clustering algorithm, created by (Dunn, 1973) and improved by (Bezdek and Bezdek, 1981), is a widely used and successful technique for data clustering analysis based on fuzzy set theory. Fuzzy C-means (FCM) clustering is an algorithm that groups data into clusters based on the degree of membership of each observation. This technique assigns a probability score to each data point for belonging to a certain cluster, allowing for overlapping clusters. With a membership degree ranging from 0 to 1, Fuzzy set theory enables data to be categorized into a set. The sum of the memberships across all clusters for a data point must equal one. Zadeh (1965) introduced the concept of uncertainty in membership of data points to clusters, which was represented by a membership function. Amongst the first to suggest the use of fuzzy set theory for cluster analysis applications were (Bellman et al., 1966) and (Ruspini, 1969). Higher degrees of membership is assigned by Fuzzy c-means to data points closer to the center of a cluster. Fuzzy c-means is based on fuzzy set theory, and it produces a clustering in which membership weights are assigned to each data point corresponding to each cluster based on the distance between the cluster center and the data point. The Fuzzy C-Means algorithm is a widely used and successful clustering technique. The degree of membership of each data point for all clusters is determined by minimizing its objective function. The function that needs to be to minimized is expressed as:

$$\mathrm{SSE}\left(C_1, C_2, \ldots, C_{\mathrm{k}}\right)=\sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{p} \operatorname{dist}\left(\mathbf{x}_i, \mathbf{c}_j\right)^2 . \tag{2.2}$$

To ensure that the clusters form a fuzzy pseudo-partition, the following conditions are imposed on the clusters, which are considered reasonable. All the weights for a given data x_i, add up to one.

$$\sum_{j=1}^{k} \omega_{ij}=1. \tag{2.3}$$

Each cluster c_j contain, with non-zero weight, at least one data, but does not contain, with a weight of one, all of the data.

Steps of Fuzzy C-Means algorithm

- A value for C that is greater than or equal to 2 is to be selected and then create an initial fuzzy pseudo-partition. Randomly assign values to all the weights (w_{ij}), ensuring that the sum of weights associated with each data point is equal to one.
  REPEAT

- Using the fuzzy pseudo-partition, compute the center of each cluster, also known as the centroid.

$$\mathrm{c}_j=\frac{\sum_{i=1}^{N} w_{ij}^{p} \cdot x_i}{\sum_{i=1}^{N} w_{ij}^{p}}. \tag{2.4}$$

- Compute the fuzzy pseudo-partition, which refers to the values of w_{ij}. The degree of membership for each data point is then adjusted (updated)

$$w_{ij}=\left(1 / \operatorname{dist}\left(\mathbf{x}_i, \mathbf{c}_j\right)^2\right)^{\frac{1}{p-1}} / \sum_{q=1}^{k}\left(1 / \operatorname{dist}\left(\mathbf{x}_i, \mathbf{c}_q\right)^2\right)^{\frac{1}{p-1}} . \tag{2.5}$$

Fuzzy C-means clustering is similar to K-means clustering when $p = 0$, but typically $p = 2$ is used (known as fuzzification) for analysis.

To arrive at the final clustering solution, the algorithm repeats steps 2 and 3 until there is no change in the centroids. Alternatively, a stopping criterion can be executed by setting a threshold for the change in the error. When the change in error falls below the specified threshold, the algorithm will terminate . Also, a threshold for the absolute change in any w_ij can also be used as a stopping criterion. After assigning random weights to each data point in the beginning, the Fuzzy C-means algorithm calculates the centroids of every cluster and the weight of each data point repeatedly until the centroid stops changing. This algorithm has a structure that is similar to the K-means algorithm, which alternates between updating the centroids and assigning each data point to its closest centroid. The objective of Fuzzy C-means is to minimize the sum of squared errors (SSE), and it behaves similarly to K-means algorithm. The weights assigned to each data point are selected randomly but must follow the condition that their sum is equal to 1.

2.4.1 Advantages of Fuzzy C-Means

Fuzzy C-Means is a clustering algorithm that assigns a degree of membership to each data point, allowing for more nuanced analysis. Fuzzy C-Means is a useful tool for clustering data, especially when dealing with complex and overlapping datasets. It can handle noisy data and overlapping clusters, making it useful in cases where data points may belong to multiple groups. FCM differs from K-means by allowing each data point to belong to multiple clusters.

2.4.2 Disadvantages of Fuzzy C-Means

Despite its advantages, Fuzzy C-Means has some downsides. Difficulty of determining the optimal number of clusters which can lead to less-than-optimal results. Another potential issue is that the algorithm may only converge to a local minimum, which can result in suboptimal clustering. Additionally, the fuzzification parameter p can be challenging to select, as it can affect the degree of fuzziness in the membership function.

2.5 Model Based Clustering

Model-based clustering is a statistical technique that can classify data points based on their probability distribution, without requiring prior knowledge of cluster shapes or assignments. This method has broad applications across several fields, including biology, finance, and social sciences, as it can reveal prominent patterns and relationships within complicated data sets. By representing data as a collection of probability distributions, each referring to a unique cluster, model-based clustering intends to determine the ideal number of clusters and their corresponding probability distributions that best describe the data. Mixture model clustering is a subtype of model-based clustering that assumes that the data arise from a combination of probability distributions, with each representing a distinct cluster. This technique is commonly employed in various fields to identify significant patterns and relationships within intricate data sets. The Gaussian Mixture Model is a widely used approach, which fits normal distributions to the dataset and is particularly effective in identifying ellipsoidal clusters (McNicholas, 2016).

Estimating mixture model parameters using the expectation maximization (EM) algorithm

Most often, we do not know which observation was generated by which distribution so we cannot straight away calculate the probability of each data point belonging to each cluster, hence, the EM algorithm is considered as the solution to this problem. At the initial state, the parameter values are guessed or chosen at random. Then the EM algorithm calculates the probability that each observation belongs to a certain distribution and then will use these probabilities to calculate a new estimate for the parameters. This process continues until the estimates of the parameters either do not change or change very little.

Steps

The steps for estimating mixture model parameters using the EM algorithm are as follows:

- Start with an initial set of model parameters. This can be done in diverge ways, such as randomly.

- In the Expectation step, calculate the probability that each observation belongs to each distribution.

- In the Maximization step, use the probabilities from the Expectation step to find new estimates of the parameters that maximize the expected likelihood.

- Repeat the Expectation and Maximization steps until the estimates of the parameters stop changing significantly. Alternatively, stop when the change in the parameters is below a specified threshold.

2.5.1 Advantages of Model Based Clustering

Mixture model clustering can identify complex and non-spherical clusters, estimate cluster probabilities, and allow for flexible cluster numbers.

2.5.2 Disadvantages of Model Based Clustering

It may not be effective when clusters can accommodate only a few observations or when data is nearly co-linear. Additionally, the choice of initial parameter values is critical and can impact the results obtained, and the method may be computationally demanding for large datasets. Further, the underlying assumptions required by the method may not always be suitable or accurate for the given dataset.

2.6 Trimmed K-means Clustering

Trimming clustering technique is a useful method for improving the accuracy and robustness of clustering results by clearing outliers and noise from a dataset before performing clustering. One specific implementation of this approach is The trimmed K-means (T-K-means) clustering algorithm. Trimmed K-means is a method introduced by (Cuesta-Albertos et al., 1997) that allows for the removal of a specified proportion of outliers during the grouping process [(García-Escudero et al., 2008), (García-Escudero et al., 2010)]. The use of this technique can enhance the quality and robustness of clustering outcomes, especially when dealing with datasets that contain noise or outliers. The process involves several steps, such as defining a dissimilarity measure, selecting a threshold for trimming, eliminating outliers, and applying a clustering algorithm to the remaining data. Trimmed K-means clustering allows for the removal of a specified percentage of outliers during the grouping process, making it a valuable tool for managing outliers within data clusters. By removing outliers, Trimmed K-means can create a new k-cluster that significantly reduces the impact of noise and outliers during cluster analysis.

Determining the parameters

To conduct Trimmed K-means clustering, users need to define two important parameters: the percentage of observations to be trimmed and the number of clusters (k) to be created. The percentage of observations to be trimmed is selected by the user and determines how many outliers and noise will be eliminated from the dataset, depending on the data characteristics and analysis goals. The number of clusters (k) is another critical parameter that should be determined before running the algorithm. In most cases, the user has to choose the value of k based on prior knowledge or using a suitable method to identify the optimal number of clusters. The choice of distance metric and initialization method used in the algorithm also affects the results and requires careful selection based on data and analysis objectives. To obtain accurate and meaningful cluster assignments, the selection of parameters is essential and necessitates thoughtful consideration based on the dataset's specific characteristics and desired analysis objectives.

2.6.1 Advantages of Trimmed K-means Clustering

Trimmed K-means is a clustering technique that can enhance the accuracy and robustness of cluster assignments by eliminating outliers and noise from the dataset, especially in datasets that exhibit high variability or noise. It also provides flexibility in the clustering process, as the percentage of observations to be trimmed can be customized based on the specific data characteristics and analysis objectives. Additionally, outlier management is another advantage of Trimmed K-means, as it allows for the management of outliers within data clusters that require grouping.

2.6.2 Advantages of Trimmed K-means Clustering

There are some drawbacks to using Trimmed K-means. Firstly, the method is restricted to the K-means clustering algorithm, making it less versatile in comparison to other clustering techniques. Secondly, the percentage of observations to be trimmed must be determined through prior knowledge of the data or by an appropriate method for establishing the optimal trimming threshold, which may not be always accessible. Moreover, removing data points could lead to the loss of essential information that is not necessarily an outlier but could significantly impact the clustering results. Lastly, depending on the extent of trimming, the resultant clusters may be less interpretable or meaningful, which could affect the overall interpretability of the analysis.

Chapter 3

Methodology

3.1 Density Power Divergence

The minimum Density Power Divergence Estimators (MDPDE), was proposed by (Basu et al., 1998). It is a method used in statistical inference to estimate the parameters of a probability density function (PDF) based on observed data. It is a variant of the power divergence estimation framework that seeks to find the parameter values that minimizes a specific divergence measure between the true unknown density function and the estimated density function. Basu et al. (1998) provide a definition of the power divergence between two densities, $f(x)$ and $g(x)$. This divergence is dependent on a single parameter, α. It is expressed as:

$$d_\alpha\left(f_\theta, g\right) = \begin{cases} \int_x \left\{ f_\theta^{1+\alpha}(x) - \left(1+\frac{1}{\alpha}\right) f_\theta^{\alpha}(x) g(x) + \frac{1}{\alpha} g^{1+\alpha}(x) \right\} dx, & \alpha > 0 \\ \int_x g(x) \log\left\{ \frac{g(x)}{f_\theta(x)} \right\} dx, & \alpha = 0. \end{cases} \tag{3.1}$$

The parameter α is a parameter that controls the trade-off between efficiency and robustness. When α is set to 0 the power divergence corresponds to the kullback-Leibler divergence and the estimation method becomes maximum likelihood estimation. When α is set to 1, the divergence becomes the mean squared error, resulting in a robust but less efficient minimum mean squared error estimation. In the analysis of the paper, our primary focus will lie on smaller values of α, approximately around 0, ranging from zero to one. However, it is also possible to consider value greater than one except that the efficiency tends to decrease as α increase.

3.1.1 The Normal Distribution

Minimum Density Power Divergence Estimate (MDPDE) can also be applied in the context of cluster analysis. In clustering, the primary objective is to group similar data together into clusters based on their shared characteristics. MDPDE leverages the concept of density power divergence, a measure of how two probability distributions differ from each other, to estimate the parameters of the underlying distribution within each cluster. In this research, the focus was on using the normal distribution. We will estimate the parameters of the normal distribution using MDPDE. This involves applying the MDPDE approach to the normal probability density function (pdf), enabling us to effectively capture the essential features of data distributions within clusters. By utilizing MDPDE with the normal distribution, we enhance our ability to achieve accurate and robust clustering outcomes. The pdf of the normal distribution is given as:

$$f_\theta(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-u}{\sigma}\right)^2\right], x \sim N\left(u, \sigma^2\right). \tag{3.2}$$

Estimation of parameters

Since the third term of divergence from equation (3.1) is independent of θ, the power divergence estimator of θ can be found by minimizing:

$$\int f_\theta^{1+\alpha}(x)dx - \left(1+\frac{1}{\alpha}\right)\frac{1}{n}\sum_{i=1}^{n} f_\theta^{(\alpha)}\left(x_i\right). \tag{3.3}$$

where

$$\int_x f_\theta^{1+\alpha}(x)dx = (2\pi)^{-\frac{\alpha}{2}}\sigma^\alpha(1+\alpha)^{-\frac{1}{2}}. \tag{3.4}$$

Equation (3.3) becomes,

$$= (2\pi)^{-\alpha/2}\sigma^{-\alpha}(1+\alpha)^{-\frac{1}{2}}\left[1-\frac{(1+\alpha)^{3/2}}{n\alpha}\sum_{i}\exp\left[-\frac{\alpha}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2\right]\right]. \tag{3.5}$$

The above equation is minimized over both the mean μ and variance σ^2. The tuning parameter α controls the trade-off between efficiency and robustness of the MDPDE - robustness measure increases if α increases, but at the same time, efficiency decreases (Das et al., 2022).

3.1.2 The Empirical DPD – Univariate Case

Our emphasis lies in estimating parameters within the normal distribution as these parameters hold significance, particularly for our chosen distance metric, the Mahalanobis distance. The estimation of parameter θ through the Minimum Density Power Divergence Estimate (MDPDE) involves minimizing the DPD measure directly, as indicated in equation (3.3). Alternatively, an iterative approach to compute MDPDE is available by solving the provided estimating equations. Suppose $X \sim N(\mu, \sigma^2)$. An iterative algorithm for the MDPDE is as follows:

$$\omega_i = \exp\left[-\frac{\alpha}{2}\left(\frac{x_i-\mu}{\sigma}\right)^2\right], \tag{3.6}$$

$$\mu = \frac{\sum_i \omega_i x_i}{\sum_i \omega_i}, \tag{3.7}$$

$$\sigma^2 = \frac{\sum_i \omega_i (x_i-u)^2}{\sum_\mathrm{i} \omega_\mathrm{i} - \frac{n\alpha}{(1+\alpha)^{3/2}}}. \tag{3.8}$$

When $\alpha = 0$, Equation (3.7) and (3.8) becomes the equation non-robust ordinary least equator.

3.1.3 The Empirical DPD – Multivariate Case

Given that real-life dataset often involve multiple variables, we shift our focus towards the multivariate case for our analysis. Suppose $X \sim N_p\left(\mu, \Sigma\right)$. The DPD measure becomes:

$$d_\alpha\left(f_\theta, g\right) = (2\pi)^{-\frac{p\alpha}{2}} |\Sigma|^{-\alpha/2} (1+\alpha)^{-\frac{1}{2}} \left[1 - \frac{(1+\alpha)^3/2}{n\alpha} \sum_i \exp\left[-\frac{\alpha}{2} B_i\right]\right], \tag{3.9}$$

where

$$B_i = \left(x_i - \mu\right)^T \Sigma^{-1} \left(x_i - \mu\right). \tag{3.10}$$

An iterative approach for the MDPDE gives:

$$u_i = \frac{\sum \omega_i x_i}{\sum \omega_i}. \tag{3.11}$$

$$\omega_i = \exp\left[-\frac{\alpha}{2} B_i\right]. \tag{3.12}$$

$$\Sigma^{-1} = \frac{\sum_i \exp\left[-\frac{\alpha}{2} B_i\right] - \frac{n\alpha}{(1+\alpha)^{3/2}}}{\sum_i \omega_i \left(x_i - \mu\right)^\top \left(x_i - \mu\right)}. \tag{3.13}$$

The algorithm mentioned above requires initial values for μ and Σ. To enhance resilience against outliers, we utilize the median for μ, and a scaled median absolute deviation (MAD) for Σ.

3.2 Robust Mahalanobis Clustering

The K-means clustering method was considered as the cluster method of which our proposed method was applied on. We considered the mahalanobis distance to group the observation. The two parameters needed in the mahalanobis distance were estimated using the MDPDE.

3.2.1 MDPDE Approach

The usual k-means clustering algorithm was adopted except that our parameters were estimated using the MDPDE. It utilizes the Iteratively Reweighted Least Squares (IRWLS) algorithm to estimate the parameters of the normal distribution. The Minimum Density Power Divergence Estimation (MDPDE) algorithm aims to find the parameter values that minimizes the DPD between the observed data and the estimated distribution. In the case of the MDPDE of a normal distribution, the algorithm updates the mean and covariance matrix iteratively using the IRWLS algorithm. The weights used in the algorithm are based on the DPD formula applied to the squared difference between the observed data and the estimated mean.

Steps

- Initialize the parameter values.
  - Choose k clusters.
  - The sample mean and covariance matrix are used as initial values, as μ and σ. The inverse covariance will be used in the analysis.
- Compute the Mahalanobis distance of each observation from the mean; and set it as B.

$$B_i = (x_i - \mu)\,\Sigma^{-1}\,(x_i - \mu)\,.$$

- Compute the weight matrix, ω.

$$\omega = \exp\left\{\frac{-\alpha * B_i}{2}\right\}.$$

- Update the center mu using the weight, ω.

$$\mu = \frac{\sum w_i x_i}{\sum w_i}.$$

- Update the inverse covariance matrix, sigma inverse using the updated mean and weight.

$$\Sigma^{-1} = \frac{\sum_i \exp\left[-\frac{\alpha}{2}B_i\right] - \frac{n\alpha}{(1+\alpha)^{3/2}}}{\sum_i \omega_i \left(x_i - \mu\right)^\top \left(x_i - \mu\right)}.$$

- Recompute the mahalanobis distance and assign each data point to the closest centroid

$$B_i = \left(x_i - \mu\right) \Sigma^{-1} \left(x_i - \mu\right).$$

- Iteratively perform steps 3,4,5, and 6 until the algorithm converges or reaches a maximum number of iterations.

Advantages of MDPDE

- **Robustness to Outliers:** MDPDE is designed to be robust in the presence of outliers, meaning it can provide reliable estimates even when the data includes unusual or erroneous values. This makes it particularly useful in situations where traditional methods might be sensitive to outliers

- **Flexibility:** MDPDE allows for different levels of sensitivity to deviations from the model, which can be adjusted by choosing an appropriate value for the power parameter.

- DPD can handle data with complex distributions and nonlinear relationships among variables.

Disadvantages of MDPDE

- **Parameter Sensitivity:** Selecting an appropriate value for this parameter can be challenging and might require domain expertise. An incorrect choice could lead to biased or inefficient results.

- **Computational Complexity:** The optimization processes involved in MDPDE calculations can be computationally intensive, particularly for large datasets or high-dimensional data.

- **Assumption of Normality:** Like many statistical methods, MDPDE assumes that data follows a certain distribution, often a normal (Gaussian) distribution. If the underlying data distribution significantly deviates from normality, MDPDE might not perform optimally and could lead to biased or inaccurate results.

3.3 Performance Metrics

n this section, we provide a summary of the internal and external evaluation metrics utilized in our study.

3.3.1 Internal Evaluation Metrics

Internal evaluation metrics for clustering are used to assess the quality of clustering results without relying on external information, such as ground truth labels. Some common internal evaluation metrics for clustering include:

- R-square (R), which is an internal evaluation metric, is a measure that compares SSE (which represents the dissimilarity between groups or clusters) to SST (which represents the total dissimilarity in the data set). It is expressed on a scale from 0 to 1, a higher R-square suggests that the clusters are well-separated and internally homogeneous (Sharma, 1995). R-square is normally affected by outliers. We therefore considered trimmed R-square as a performance metrics. Trimmed R-square measures the proportion of the variance in the data that is explained by the clustering algorithm while excluding a certain percentage of the extreme observations.

Mathematically, it is defined as:

$$R^2 = 1 - \frac{SSE}{SST}, \tag{3.14}$$

$$TR^2 = 1 - \frac{SSE_{trimmed}}{SST_{trimmed}}. \tag{3.15}$$

- The silhouette index (SI) is a measure used in cluster analysis to assess the quality of clustering. It quantifies how similar an object is to its own cluster compared to other clusters. The silhouette index for a data point x_i in a clustering result is calculated as follows:

$$S(x_i) = \frac{b(x_i) - a(x_i)}{max\{b(x_i), a(x_i)\}}, \tag{3.16}$$

$$SI = \frac{1}{n} \sum_{i=1}^{n} S\left(x_i\right). \tag{3.17}$$

The range of silhouette index is [-1, +1]. A higher silhouette index depicts better clustering algorithm. The challenge with silhouette index is that, the value is generally high for convex clusters than other concepts of clusters (Jumadi Dehotman Sitompul et al., 2019)

- Davies-Bouldin Index (DBI) is an internal evaluation metric which measures the average similarity between clusters and their most similar clusters, and it produces a score between zero and infinity, where a lower score means better clustering performance. It works better on convex clusters such as DBSCAN. It is calculated as:

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d\left(c_i, c_j\right)} \right). \tag{3.18}$$

σ_i is the average distance of all elements in cluster i to centroid c_i.

- Calinski-Harabasz Index (CHI) measures the ratio between the dispersion within clusters and the dispersion between clusters, and a higher score indicates better clustering outcomes. The Calinski-Harabasz Index normally works better on convex clusters. Mathematically expressed as:

$$CH = \frac{\mathrm{tr}\left(B_k\right)}{\mathrm{tr}\left(W_k\right)} \times \frac{n_E - k}{k-1} \tag{3.19}$$

where $\mathrm{tr}\left(B_k\right)$ is trace of the between group dispersion matrix and $\mathrm{tr}\left(W_k\right)$ is the trace of the within-cluster dispersion matrix (Liu, 2022).

3.3.2 External Evaluation Metrics

External evaluation metrics assess the quality of clustering results by comparing them with external information or ground truth labels. Some external evaluation metrics for clustering include accuracy and kappa. Although these measures are generally used in classification problems, we adopted them in our analysis since the true class labels are availabale in each dataset.

Table 3.1: The 2x2 table for binary classification.

		Cluster labels	
		Positive (P)	**Negative (N)**
True class labels	**Positive (P)**	n_1	n_2
	Negative (N)	n_3	n_4

- Accuracy: Accuracy measures the level of agreement between the cluster labels and the class labels. The accuracy is calculated as:

  $$Accuracy = \frac{n_1 + n_4}{n_1 + n_2 + n_3 + n_4.}$$

  Higher accuracy values indicate better clustering performance (Ahmed et al., 2020).

- Kappa: the Cohen's kappa coefficient measures the agreement between the clustering results and the ground truth. Kappa ranges from -1 to 1, where +1 indicates agreement between the clustering and the ground truth, 0 indicates agreement by chance, and negative values indicate agreement worse than chance:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}, \tag{3.20}$$

$$P(A) = \frac{n_1 + n_4}{n_1 + n_2 + n_3 + n_4}, \tag{3.21}$$

$$P(E) = \frac{(n_1 + n_2) \times (n_1 + n_3) + (n_3 + n_4) \times (n_2 + n_4)}{(n_1 + n_2 + n_3 + n_4)}. \tag{3.22}$$

Chapter 4

Analysis and Results

4.1 Simulation Study

We perform a simulation experiment to validate the performance of the developed clustering algorithm against some of the existing methods based their plots and some performance metric.

Dataset

The datasets that are used to experiment the various clustering algorithms are generated from Mass, Mlbench and factoextra packages. The names of the datasets are below;

- Dataset with outliers
- Dataset with different means
- Spherical Dataset
- Multishape Dataset

Performance Metrics

Different performance metrics are used to show the performance of different algorithms on different datasets. However, the emphasis was placed on accuracy and kappa in evaluating the best clustering algorithm as all the internal algorithm works perfectly on density based algorithms.

- Accuracy; the value ranges from 0 to 1. The value 1 indicates that the clustering algorithm has produced results that closely match the ground truth, suggesting a strong performance, where zero suggest poor performance.

- Kappa is a metric that varies between 0 and 1. A score of 1 signifies that the clustering algorithm has achieved results that closely align with the ground truth, indicating a robust performance. Conversely, a score of 0 indicates poor performance, where the clustering results significantly deviate from the ground truth

- R-square value ranges from 0 to 1. We will consider the trimmed R-square (TR^2) as the performance metric in evaluating the cluster methods. A value of 0 indicates that the clustering algorithms didn't perform to the expectation. A value of 1 indicates that the clustering algorithms performed very well.

Clustering Algorithm

An α value of 0.2 was considered as the DDP parameter for the developed method. The developed algorithm require the number of clusters needed to be generated, and that is k. We named the developed algorithm as Mahalanobis DPD (Maha. DPD); and considered how the results will be after the first iteration, thereby naming that one as Starter DPD. Six known clustering methods were considered. k-means using euclidean distance and k-means using mahalanobis distance. Both algorithms behaves the same when there is no correlation in the dataset. DBSCAN algorithm was considered and the two parameters (ϵ and Minpts) needed in the algorithm were provided for each instances. Fuzzy C-means, Gaussian Mixture Model required only the number of clusters as the input, and from each dataset, k was provided. Trimmed k-means is the last algorithm which was considered in the analysis as an existing method. 1% of the datasets were trimmed as most of the simulation has little outliers. K, the number of clusters, was provided.

4.1.1 Experiment on Dataset with Outliers

The simulations was conducted in the R statistical software, utilizing the MASS package. This package facilitated the generation of mixture models with multivariate normal component distributions, from which observations were drawn. The data contains 500 samples, 2 features and known classification. Three groups are defined, in this simulation, each with their respective means and covariance matrices. Group 1 has a mean vector of (5, 7) and a covariance matrix that is double the identity matrix with off-diagonal elements of 0.5. Group 2 has a mean vector of (10, 5) and a covariance matrix with off-diagonal elements of -0.5. Group 3 has a mean vector of $(-1, 10)$ and a covariance matrix that is twice the identity matrix. Outliers were introduced into the dataset. These outliers are appended to the existing dataset x, which resulted in a dataset that includes the original clusters along with the introduced outliers. The developed clustering algorithms (Mahalanobis DPD) were compared to Standard K-means, Mahalanobis K-means, DBSCAN, Fuzzy C-Means(FCM), Gaussian Mixture Model (GMM) and, Trimmed K-means (T-K-means). The study considered different parameter settings for clusters generated to assess the performance of each algorithm. The algorithms disregarded classification information and instead proceeded to assign observations into clusters. The primary focus of the simulation study was to compare the performance of each algorithm in the presence of outliers. The result are shown in the figure below:

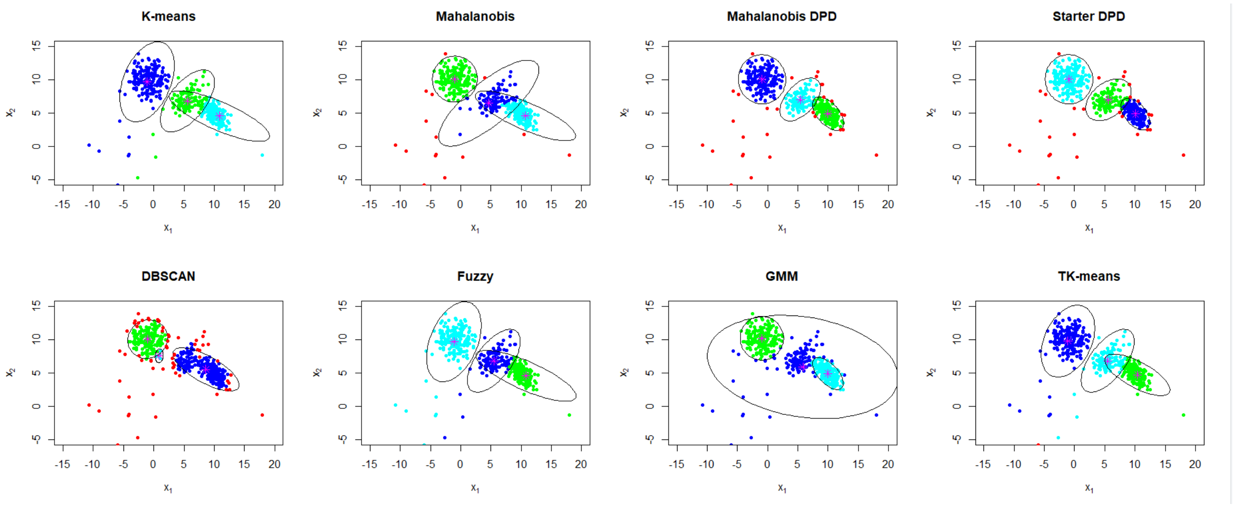


Figure 4.1: Cluster plots for simulated dataset with outliers.

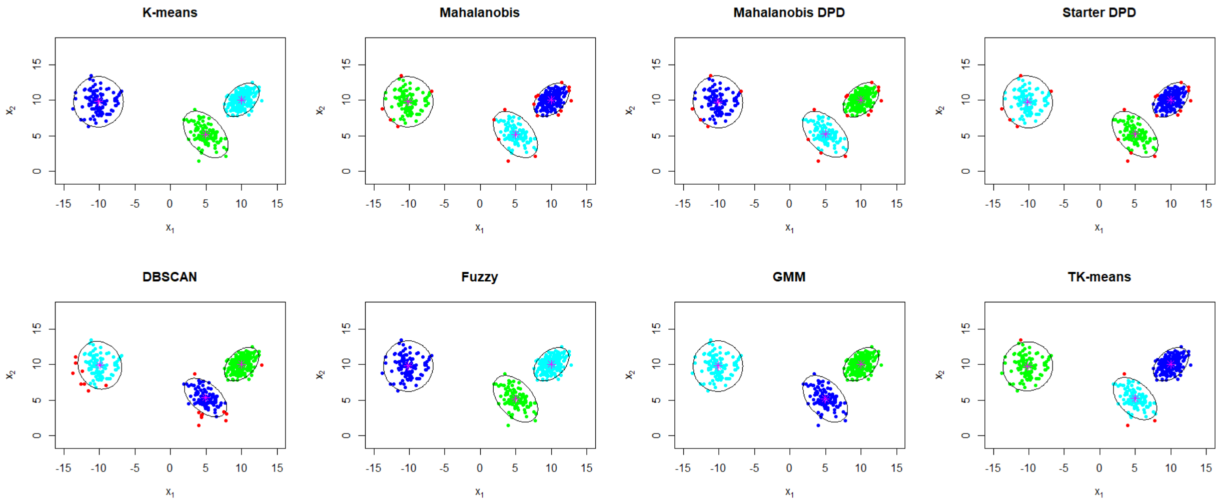
Table 4.1: Performance measures (PE) of simulated dataset with outliers.

PM Method	Accuracy	Kappa	TR^2	R^2	SI	DBI	CHI	cluster	outlier
K-means	0.935	0.901	0.892	0.759	0.545	0.995	812	3	0
M-K-means	0.938	0.907	0.894	0.741	0.550	1.155	741	3	24
Maha. DPD	**0.956**	**0.932**	**0.909**	0.724	0.549	1.122	679	3	39
Starter DPD	0.938	0.907	0.908	0.715	0.549	1.214	649	3	38
DBSCAN	0.738	0.595	0.833	–	0.296	**0.613**	**2206**	3	75
FCM	0.940	0.909	0.893	0.758	0.544	0.992	811	3	0
GMM	0.923	0.884	0.771	0.641	0.530	1.725	462	3	0
T-K-means	0.952	0.927	0.811	–	**0.563**	0.894	1322	3	6

To compare the newly developed algorithm with the existing ones, the cluster classifications obtained were placed side by side with the known cluster classifications. The proportion of correctly classified observations was then calculated for each algorithm. From Figure 4.2, we clearly see that the Mahalanobis DPD shows what is graphically best. Table 4.6 confirms that indeed our developed algorithm outperformed the other clustering method, as it can be seen that the accuracy and Kappa values for the Mahalanobis DPD is greater than all the other methods, recording an accuracy value of 0.958, with Trimmed K-means scoring 0.952. Our developed algorithm and the trimmed k-means performed very well because they are robust in nature and insensitive to outliers hence they are able to match the clusters unto their true classification clusters even in the present of outliers. The trimmed r-square value shows that the performance of all algorithm is good as their values are greater than 0.5. DBI depicts that DBSCAN is the best among all algorithm, and this is because, DBI is better on density based algorithms. However considering the cluster plot and the performance measures, it is clear that DBSCAN performed worst on this dataset.

4.1.2 Experiment on Dataset with Different Means

Dataset with different were generated in order to know the performance of each clustering algorithm. The dataset contains 400 samples, 2 features with known classification. Three clusters with different means are defined in this simulation using the ‘mvrnorm’ function in R. Group 1 contains 100 data points with mean vector set to (5, 5). Group 2 contains data points with mean vector set to (10, 10). Group 3 contains 100 data points with mean vector is set to $(-10, 10)$. The data points from all three groups were combined into a single dataset called ‘x’ using the ‘rbind’ function. The existing methods and the developed algorithm were applied to the dataset. The result can be seen in the figure below.


K-means

Mahalanobis

Mahalanobis DPD

Starter DPD

DBSCAN

Fuzzy

GMM

TK-means

Figure 4.2: Cluster plots for simulated dataset with different means.

Table 4.2: Performance measures (PE) of simulated dataset with different means.

PM **Method**	**Accuracy**	**Kappa**	$\mathbf{TR^2}$	$\mathbf{R^2}$	**SI**	**DBI**	**CHI**	**cluster**	**outlier**
K-means	**1.000**	**1.000**	0.960	0.962	**0.756**	0.389	5062	3	0
M-K-means	**1.000**	**1.000**	**0.967**	0.962	**0.756**	0.389	5062	3	20
Maha. DPD	**1.000**	**1.000**	**0.967**	0.962	**0.756**	0.389	5062	3	19
Starter DPD	**1.000**	**1.000**	**0.967**	0.962	**0.756**	0.389	5062	3	19
DBSCAN	0.955	0.929	**0.966**	-	0.684	**0.366**	**6162**	3	18
FCM	**1.000**	**1.000**	0.960	0.962	**0.756**	0.389	5062	3	0
GMM	**1.000**	**1.000**	0.960	0.962	**0.756**	0.389	5062	3	0
T-K-means	0.990	0.984	0.964	-	0.736	0.377	5369	3	4

From figure 2, it can be seen that all algorithms look what is visually best with most algorithm detecting outliers except algorithms like K-means, Fuzzy C-means and Gaussian Mixture model which are unable to detect outliers. The trimmed r-square values show that all algorithms indeed performed very well on this dataset. All the algorithm also have a silhoutte index which is greater greater than 0.5. Davies-Bouldin Index drastically decreased on this dataset, showing that indeed all algorithms perform very well. From the accuracy and kappa, we clearly see the best algorithms for this dataset. As k-means, Mahalanobis k-means, Mahalanobis DPD, Starter DPD, Fuzzy C-means and Gaussian Mixture model shows a convincing score considering their accuracy and kappa values, depicting that they did a perfect matching comparing cluster labels and true class labels. All algorithm produced three clusters with mahalanobis k-means detecting the highest outliers, 20.

4.1.3 Experiment on Spherical Dataset

Mlbench package was used to generate the sperical dataset. The dataset contains 1000 samples with two features and classes. The number of data points were specified as 1000 and the outcome of the ‘spherical data’ contains the coordinates of points in a spiral pattern. To standardize the spherical data, ‘scale’ function was used to scale the data, ensuring that each variable has a mean of 0 and a standard deviation of 1. The developed algorithm and the existing clustering methods were applied on this dataset. The result is shown below.

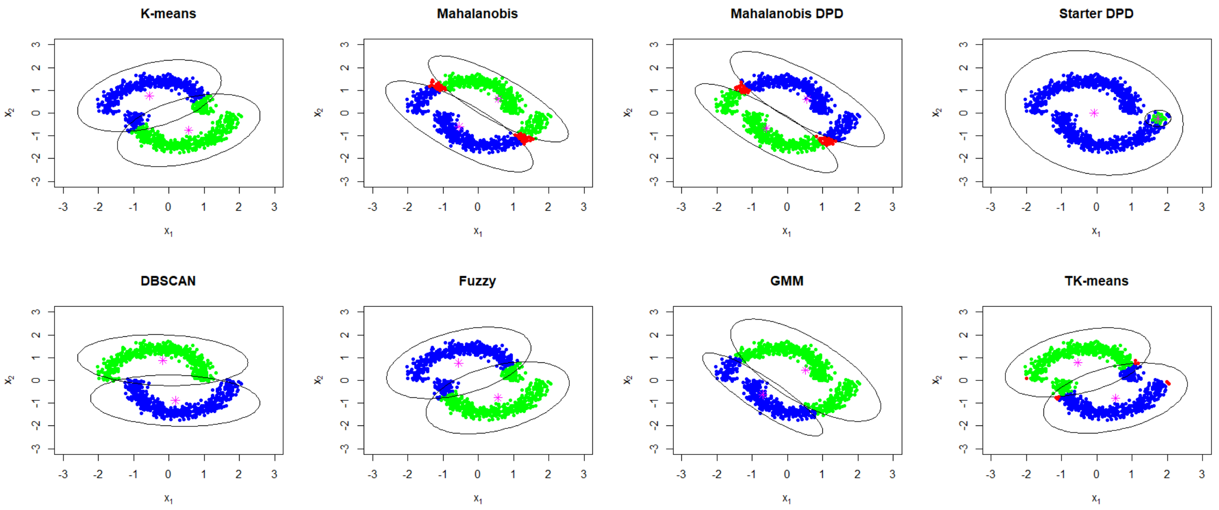


Figure 4.3: Cluster plots for spherical dataset.

Table 4.3: Performance measures (PE) of simulated spherical dataset.

PM Method	Accuracy	Kappa	TR^2	R^2	SI	DBI	CHI	cluster	outlier
K-means	0.802	0.604	0.463	0.446	0.401	1.115	803	2	0
MK-means	0.766	0.532	0.376	0.336	0.376	1.405	505	2	49
Maha. DPD	0.763	0.526	0.379	0.334	0.374	1.406	499	2	55
Starter DPD	0.531	0.062	0.049	0.049	0.100	**0.859**	52	2	0
DBSCAN	**1.000**	**1.000**	0.451	-	**0.405**	1.198	695	2	0
FCM	0.804	0.608	**0.463**	0.446	0.402	1.115	802	2	0
GMM	0.736	0.472	0.357	0.321	0.362	1.407	470	2	0
T-k-means	0.806	0.616	0.452	-	0.314	1.102	**846**	2	10

From figure 4.3, we clearly see DBSCAN shows what is visually best. This can be confirmed with the accuracy and kappa values. DBSCAN is actually known to cluster arbitrary shapes and sizes and this is the reason why it outperforms all the other algorithms. DBSCAN has an accuracy value of 100% and kappa value of 100%, which is the highest amongst all algorithms, signifying a better match with the known classification. The trimmed K-means algorithm is the second best using the accuracy and kappa values. The silhoutte index shows that DBSCAN indeed outperforms the other algorithm, as it has higher value than all algorithm. It can be seen from almost all the performance measures that our our developed algorithm performs poorly on such dataset, as it fails to cluster this dataset. The starter DPD is the worst among all algorithms.

4.1.4 Experiment on Multishape Dataset

In this experiment, ‘factoextra’ package was used to generate the Multishape dataset. This dataset contains multiple shapes that can be used for various analysis purposes. The dataset contains 1100 samples with two features and their classes. The clustering algorithms have been used on this dataset and the results can be seen below:

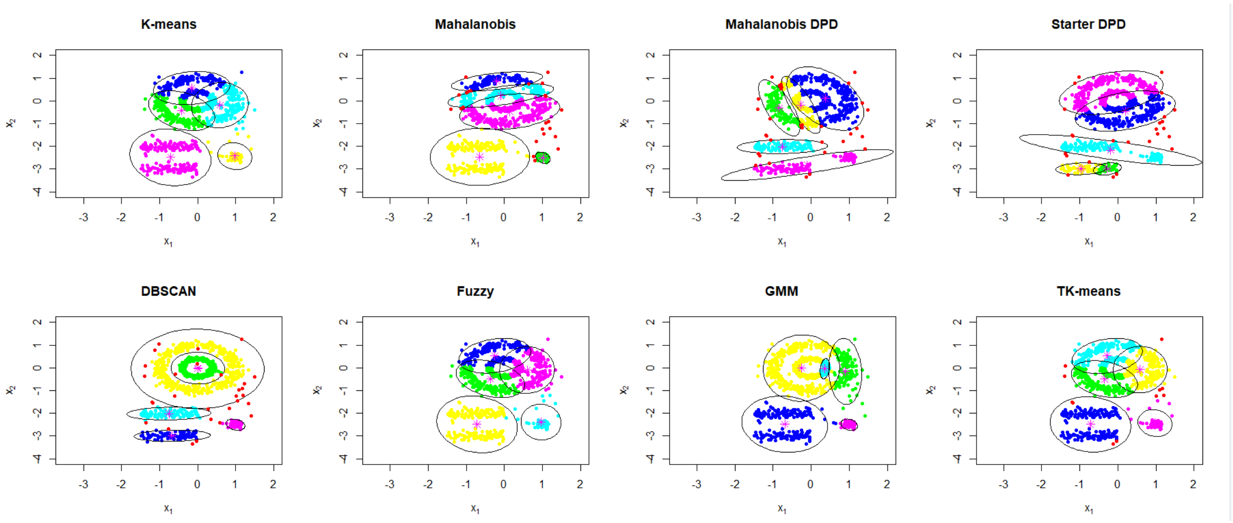


Figure 4.4: Cluster plots for simulated multishape dataset.

Table 4.4: Performance measures (PE) of simulated multishape dataset.

PM Method	Accuracy	Kappa	TR^2	R^2	SI	DBI	CHI	cluster	outlier
K-means	0.392	0.227	0.823	0.842	**0.410**	0.817	1461	5	0
M-K- means	0.503	0.326	0.799	0.783	0.311	1.279	990	5	24
Maha. DPD	0.538	0.369	0.765	0.756	0.246	1.649	849	5	33
Starter DPD	0.512	0.306	0.756	0.750	0.302	1.303	821	5	25
DBSCAN	**0.974**	**0.963**	0.723	-	0.241	19.734	755	5	29
FCM	0.390	0.224	0.823	0.842	**0.410**	0.835	1461	5	0
GMM	0.515	0.341	0.738	0.764	0.190	1.095	886	5	0
T-K-means	0.392	0.234	**0.847**	-	0.407	**0.806**	**1544**	5	11

DBSCAN is also known to cluster mulitishape and complex dataset; and once again, it can be seen from figure 4.4 that it looks what is visually best. It has the highest accuracy value of 97.4% and a kappa value of 96.3%, which indicates that it indeed outperformed all the other algorithms. From table 4.4, Our developed algorithm, Mahalanobis DPD wasn't that bad as compared to the remaining algorithm. Indeed, it is the second best algorithm considering the accuracy and kappa values. Mahalanobis DPD clustered 53.8% of the entire dataset correctly. With a kappa value of 0.369. Trimmed k-means recorded the highest r-square as it trimmed a portion of the dataset as outliers, making it suitable using the trimmed r-square. The Silhoutte index shows that k-means and Fuzzy C-means performs better than all algorithm. DBI shows that T-K-means is the best amongst all algorithms. Graphically, DBSCAN and our developed method, mahalanobis DPD performed better than other algorithms on this dataset, and this is confirmed with the accuracy and kappa values for each algorithm on this dataset, as they both indicates the percentage of data points that were correctly clustered.

4.2 Real Data Analysis

Data Source and Description

To show that the modified algorithm works, we considered two real life datasets; namely, iris flower dataset and wheat seed dataset.

Iris flower dataset

The Iris dataset was obtained from Scikit-learn, a machine learning library, where the dataset is readily available. The Iris dataset which is one of the earliest and most widely used datasets in data mining comprises 150 samples and is organized into three distinct classes: Setosa, Versicolor, and Virginica. Each class contains 50 sample data points. The dataset consists of four numeric attributes representing measurements in centimeters: Sepal length, Sepal width, Petal length, and Petal width. The fifth attribute is qualitative, indicating the class name corresponding to the plant species. The goal is to assess and compare the performance of the modified method, Mahalanobis DPD and some existing clustering methods and assess their robustness and effectiveness in the presence of outliers.

Wheat seed dataset

The Wheat Seeds dataset, sourced from the UCI Machine Learning Repository, contains data related to the geometric parameters of wheat kernels from three different varieties: Kama, Rosa, and Canadian. The dataset comprises 210 instances and 7 numeric attributes, representing the area, perimeter, compactness, length, width, asymmetry coefficient, and length of the kernel groove of the wheat seeds. The data type for this dataset is numeric, specifically floats (real numbers). The objective is to apply unsupervised learning, particularly cluster analysis to explore inherent relationships between the physical attributes of wheat seeds and their corresponding wheat variety. The dataset includes 70 randomly selected elements from each wheat variety, making it suitable for cluster analysis to identify similarities and patterns among wheat samples based on their measured characteristics.

The target variable, which is the type of wheat (Kama, Rosa, or Canadian) will be excluded from the clustering analysis.

4.2.1 Experiment on Iris Flower Dataset

We will assess the performance of the modified algorithm and the existing clustering methods based on visual inspection and performance metrics. The number of clusters, k and DPD parameter, α, were the two parameters which were need in our modified algorithm, Mahalanobis DPD. The number of clusters, k, was chosen from the known classes in the dataset. The DPD parameter α, considered for this dataset was 0.2. For now, we did not consider how to chose optimum α for the analysis, we leave the investigation for future work. We consider the performance of each algorithm. We focused on two performance metrics accuracy, and kappa to determine the best performing algorithm, as the remaining metrics seems biased to certain clustering algorithm. Accuracy and kappa both finds the best match between the class labels and cluster labels showing which cluster algorithm performed very well in matching the class labels to clusters labels. We presented some internal performance metrics however they do not share a common view on what is a good clustering. They present strong biased that do not necessarily indicate good clusters. Hence, we opted for external performance such as accuracy and kappa to conclude in the best algorithm as the class labels were already available in this dataset.

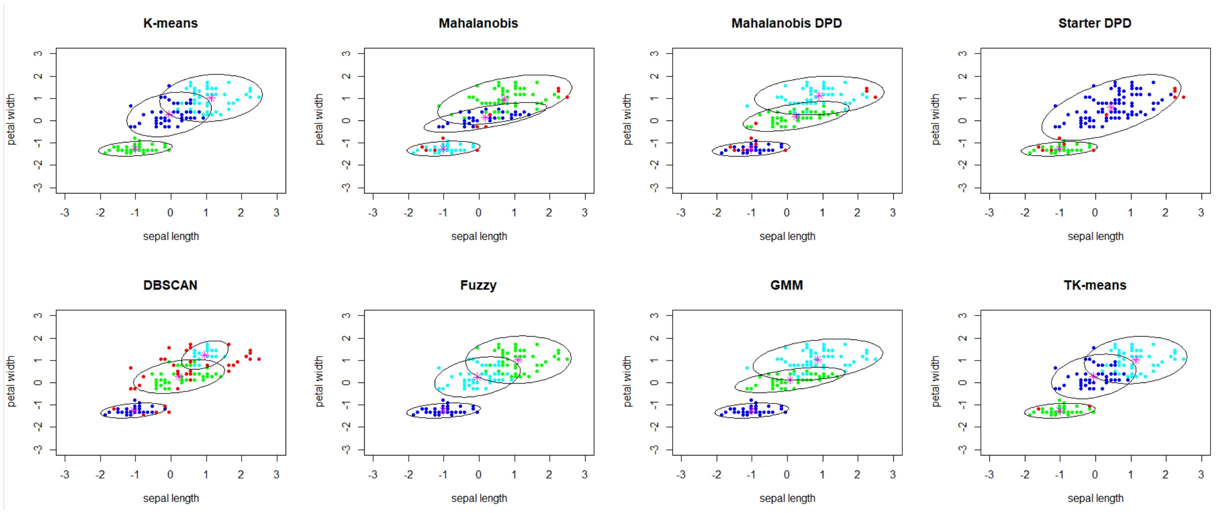


Figure 4.5: Cluster plots using sepal length and petal width.

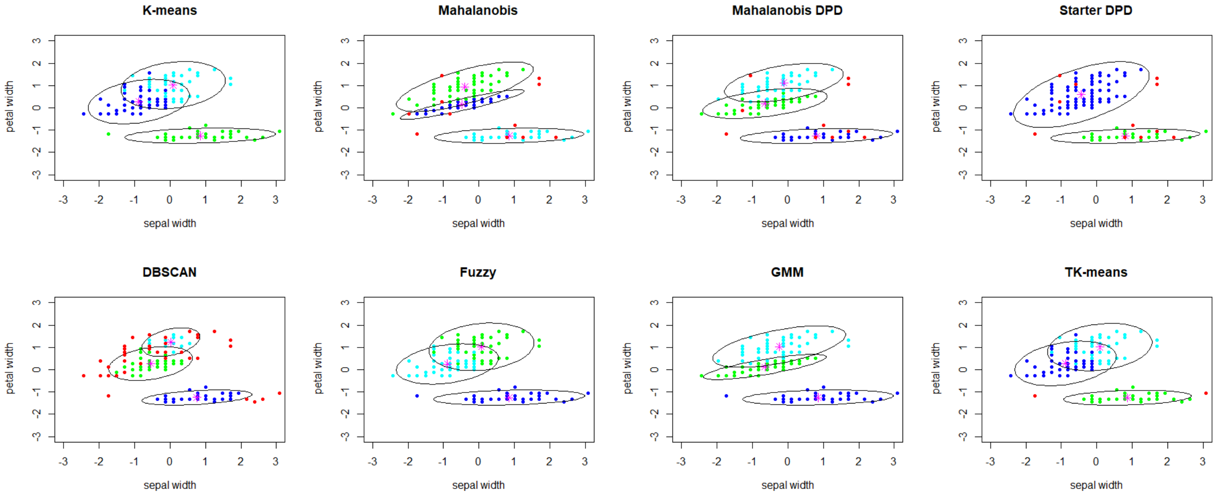


Figure 4.6: Cluster plots using sepal width and petal width.

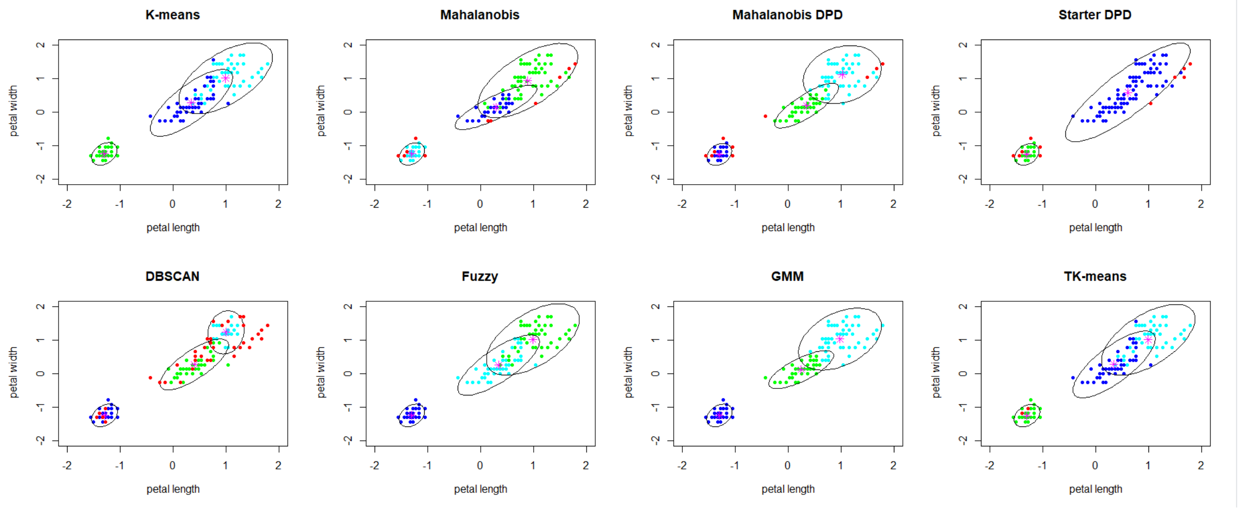


Figure 4.7: Cluster plots using petal length and petal width.

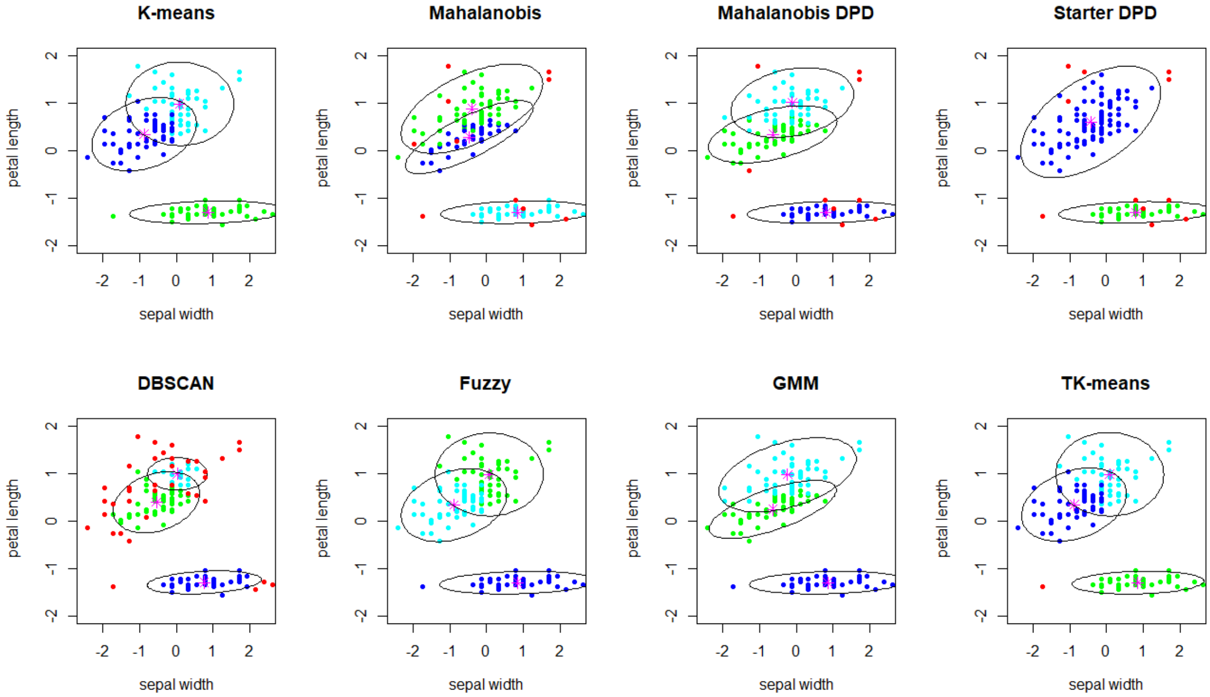


Figure 4.8: Cluster plots using sepal width and petal length.

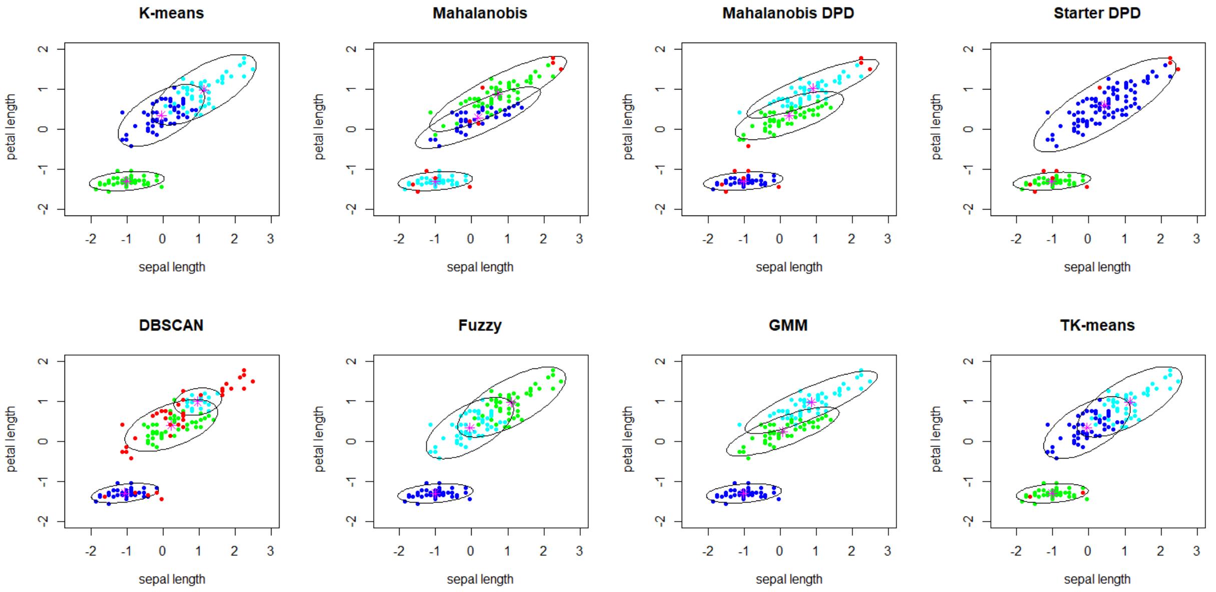


Figure 4.9: Cluster plots using sepal length and petal length.

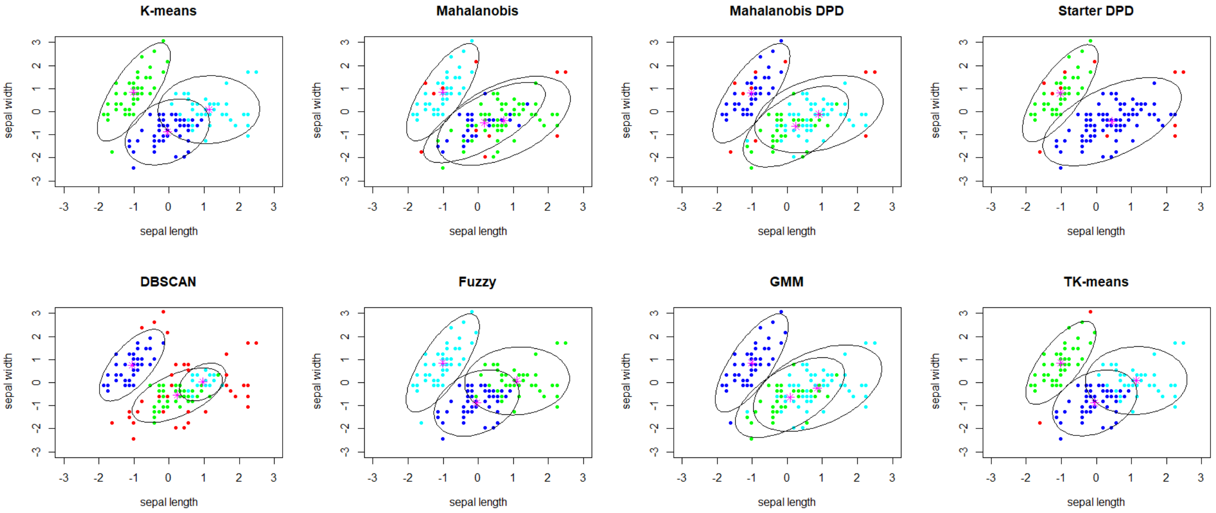


Figure 4.10: Cluster plots using sepal length and sepal width.

Table 4.5: Performance measures (PE) of iris flower dataset.

PM **Method**	**Accuracy**	**Kappa**	**TR^2**	**R^2**	**SI**	**DBI**	**CHI**	**cluster**	**outlier**
K-means	0.833	0.750	0.821	0.767	0.460	0.914	241	3	0
M-K-means	0.913	0.870	0.707	0.681	0.326	1.631	156	3	11
Maha. DPD	**0.987**	**0.980**	0.736	0.722	0.378	1.209	190	3	10
Starter DPD	0.667	0.500	0.645	0.629	**0.582**	**0.683**	251	3	11
DBSCAN	0.700	0.577	**0.830**	-	0.257	0.781	**640**	3	39
FCM	0.840	0.760	0.819	0.767	0.458	0.915	241	3	0
GMM	0.967	0.950	0.793	0.719	0.374	1.224	188	3	0
T-K-means	0.820	0.732	0.781	-	0.429	0.900	274	3	2

From figure 4.5, our modified method, mahalanobis DPD and Gaussian mixture model seems what is visually best. And this can be confirmed through their accuracy and kappa values. Our modified algorithm, mahalanobis DPD grouped 97.3% of the entire dataset correctly, which is the highest accuracy as compared to all other algorithms, depicting that our modified algorithm outperformed all the other clustering methods in terms of overall accuracy. Considering the kappa values for all algorithms, we can see that our modified algorithm recorded the highest kappa value of 0.980, depicting that Mahalanobis DPD performs better than existing algorithms as well as the starter DPD. Gaussian mixture model is the second best algorithm, recording an accuracy value of 0.967 and a kappa value of 0.950. The starter DPD performed worst as compared to all algorithm as it recorded the lowest accuracy and kappa values. DBSCAN detected the highest outliers compared to all other methods with each clustering algorithm producing three clusters in their clustering results as shown in table 4.5, however failed to correctly cluster the dataset to its true classes.

4.2.2 Experiment on Wheat Seed Dataset

In this subsection, we further verify the performance of the developed method and compar it with other clustering methods, such as K-means, Fuzzy C-means, DBSCAN, Trimmed K-means and Gaussian mixture model on another real world dataset called wheat seed dataset. Performance metrics are used to measure the quality of the various clustering results. Accuracy and kappa were adopted as the main performance metrics to assess the various clustering results.

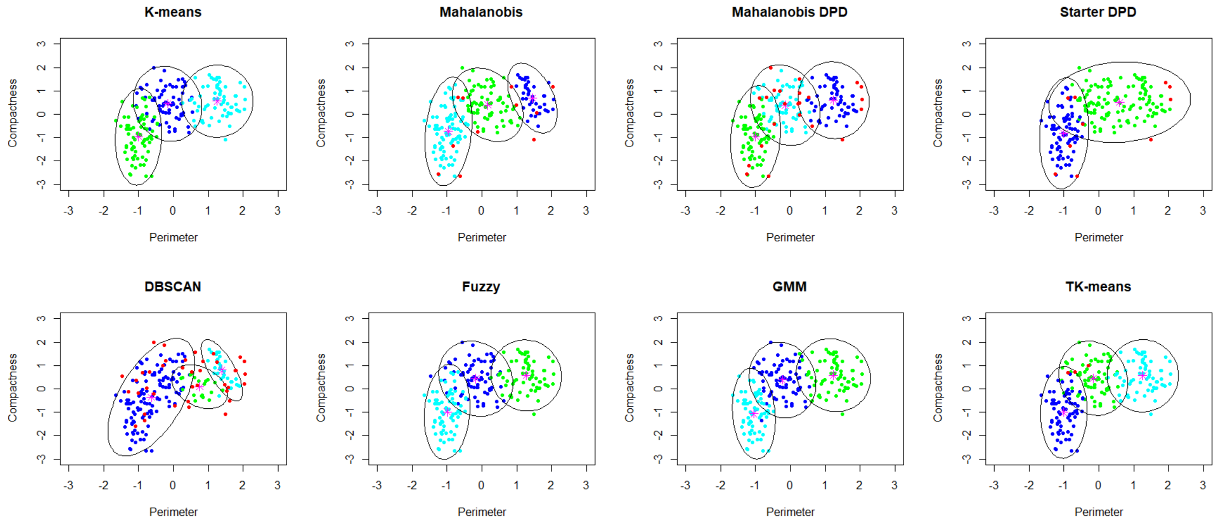


Figure 4.11: Cluster plots using perimeter and compactness.

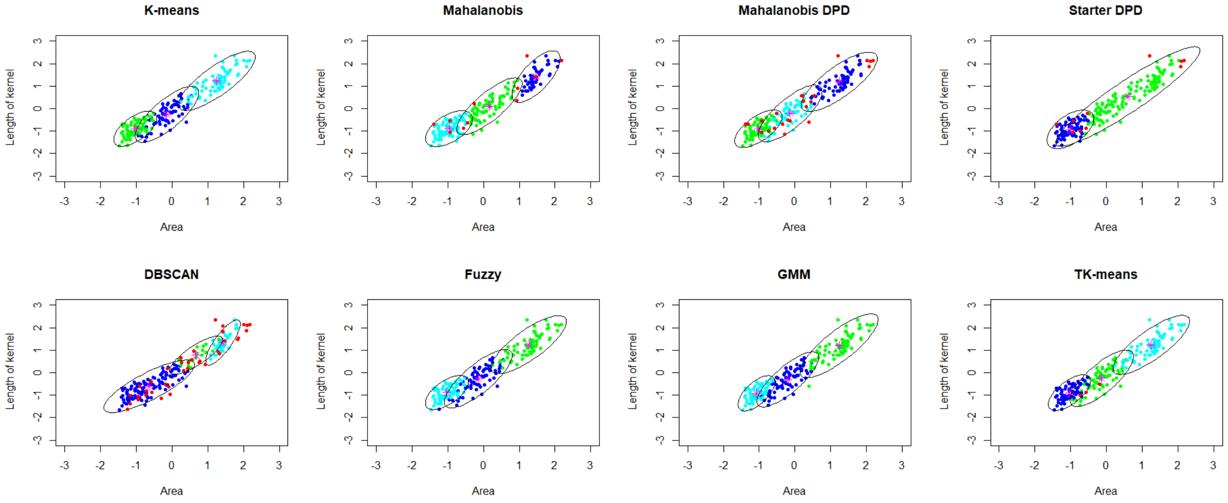


Figure 4.12: Cluster plots using area and length of kernel.

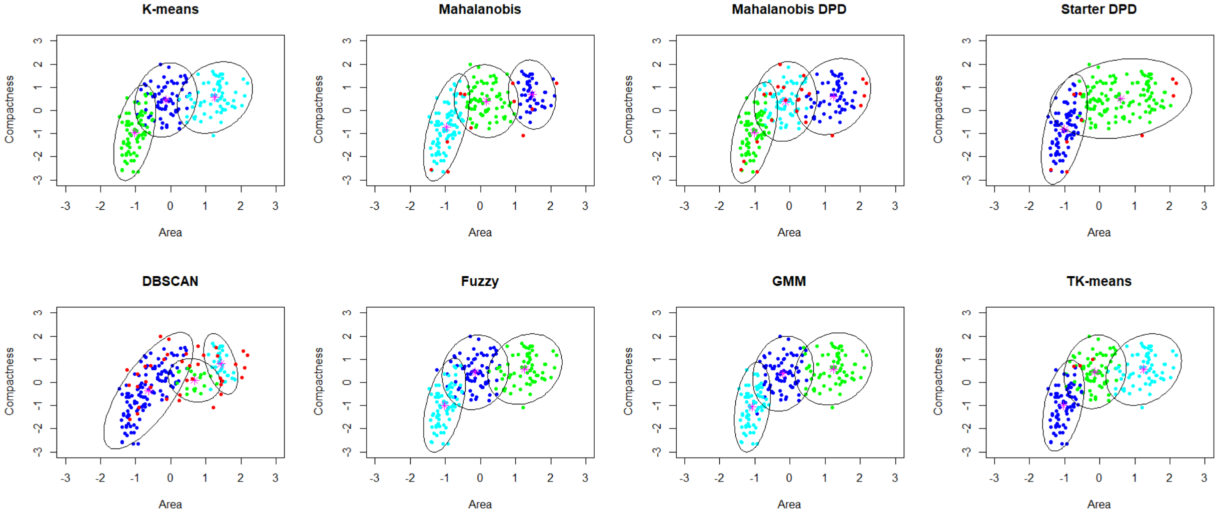


Figure 4.13: Cluster plots using area and compactness.

Table 4.6: Performance measures(PE) of wheat seed dataset.

PM **Method**	**Accuracy**	**Kappa**	**TR^2**	**R^2**	**SI**	**DBI**	**CHI**	**cluster**	**outlier**
K-means	0.919	0.879	**0.762**	0.707	**0.401**	1.005	249	3	0
M-K-means	0.819	0.729	0.683	0.670	0.350	1.086	210	3	11
Maha. DPD	**0.933**	**0.900**	0.739	0.700	0.391	1.026	241	3	23
Starter DPD	0.662	0.493	0.482	0.467	0.387	0.986	182	3	10
DBSCAN	0.548	0.351	0.590	-	0.200	0.930	237	3	39
FCM	0.919	0.879	0.760	0.707	**0.401**	**0.008**	249	3	0
GMM	0.895	0.843	0.756	0.685	0.373	1.095	225	3	0
T-K-means	0.900	0.851	0.720	-	0.388	0.969	**273**	3	3

The clustering results of the wheat dataset is displayed in figure 4.6. All the algorithm produces result that seems to be overlapping. Our developed algorithm is able to detect 23 outliers where as, DBSCAN detected the highest outliers, 39. DBSCAN has the worst clustering performance. K-means, Fuzzy C-means and GMM are unable to identify outliers. We will access the best algorithm using the performance measures. From table 4.6, our developed method has the highest accuracy, clustering 93.3% of the dataset correctly. With, K-means and Fuzzy C-means recording the second highest, with an accuracy of 0.919. The kappa value for our developed method is also the highest as compared to all the algorithm, recording a value of 0.9 out of 1. DBSCAN has the lowest value for both accuracy and kappa, indicating that it performs worst than any other algorithm on the wheat dataset. Clearly, our developed method outperforms other algorithm in the presence of outliers.

Chapter 5

Conclusion

This study enhanced the K-means clustering method by incorporating Density Power Divergence (DPD) for robust mean and covariance estimation essential for effective observation clustering. While K-means clustering is renowned for its computational speed and accuracy in many cases, it falls short when confronted with datasets that contain outliers. Our modified clustering approach, named Mahalanobis DPD, tackles this challenge by leveraging Density Power Divergence (DPD) to estimate cluster means and covariances. This adaptation enhances its resistance to outliers, countering the potential distortion of K-means results. In the comparative analysis against various clustering methods, both on simulated and real-world datasets, Mahalanobis DPD consistently emerged as the superior performer, particularly on datasets featuring outliers, provided the clusters are from multivariate normal distributions. For instance, on a simulated dataset containing outliers, Mahalanobis DPD demonstrated an impressive accuracy of 0.958 with a kappa value of 0.932. The efficacy of Mahalanobis DPD also extended to real-world datasets, with accuracy rates of 97.3% on the Iris dataset and 93.3% on the wheat seed dataset. These outcomes underline Mahalanobis DPD's promise as a practical clustering solution for real-world scenarios. It's essential to acknowledge that Mahalanobis DPD may not consistently outperform other methods depending on the dataset and the shape. However, its potential shines particularly when dealing with outlier-inclusive datasets. In conclusion, our research underscores the efficacy of Mahalanobis DPD for clustering data containing outliers. This method offers enhanced resilience against outliers compared to traditional K-means, leading to improved accuracy on outlier-prone datasets. With its adaptability across various dataset types, Mahalanobis DPD presents a promising avenue for diverse clustering applications.

Acknowledgment: I would like to express my sincere gratitude to the collaborators of Dr. Abhijit Mandal for providing me with their invaluable research proposal, which served as the foundation for my thesis. Their contribution to developing the clustering algorithms (based on the DPD measure and Mahalanobis distance) and R codes was critical to the success of this project. I do not claim any copywriting for these methods and the corresponding codes, as it was solely created by Dr. Mandal and his team.

Bibliography

Aggarwal, C. C. et al. (2015), *Data mining: the textbook*, Vol. 1, Springer.

Ahmed, M., Seraj, R. and Islam, S. M. S. (2020), 'The k-means algorithm: A comprehensive survey and performance evaluation', *Electronics* **9**(8), 1295.

Anum, A. T. and Pokojovy, M. (2023), 'A hybrid method for density power divergence minimization with application to robust univariate location and scale estimation', *Communications in Statistics-Theory and Methods* pp. 1–24.

Banerjee, A. and Davé, R. N. (2012), 'Robust clustering', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2**(1), 29–59.

Basu, A., Harris, I. R., Hjort, N. L. and Jones, M. (1998), 'Robust and efficient estimation by minimising a density power divergence', *Biometrika* **85**(3), 549–559.

Bellman, R., Kalaba, R. and Zadeh, L. (1966), 'Abstraction and pattern classification', *Journal of Mathematical Analysis and Applications* **13**(1), 1–7.

Bezdek, J. C. and Bezdek, J. C. (1981), 'Objective function clustering', *Pattern recognition with fuzzy objective function algorithms* pp. 43–93.

Cuesta-Albertos, J. A., Gordaliza, A. and Matrán, C. (1997), 'Trimmed k-means: an attempt to robustify quantizers', *The Annals of Statistics* **25**(2), 553–576.

Das, J., Beyaztas, B. H., Mac-Ocloo, M. K., Majumdar, A. and Mandal, A. (2022), 'Testing equality of multiple population means under contaminated normal model using the density power divergence', *Entropy* **24**(9), 1189.

Davé, R. N. and Krishnapuram, R. (1997), 'Robust clustering methods: a unified view', *IEEE Transactions on fuzzy systems* **5**(2), 270–293.

Dunn, J. C. (1973), 'A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters'.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise., *in* 'kdd', Vol. 96, pp. 226–231.

García-Escudero, L. A., Gordaliza, A., Matrán, C. and Mayo-Iscar, A. (2008), 'A general trimming approach to robust cluster analysis'.

García-Escudero, L. A., Gordaliza, A., Matrán, C. and Mayo-Iscar, A. (2010), 'A review of robust clustering methods', *Advances in Data Analysis and Classification* **4**, 89–109.

Johnson, R. A., Wichern, D. W. et al. (2002), 'Applied multivariate statistical analysis'.

Jumadi Dehotman Sitompul, B., Salim Sitompul, O. and Sihombing, P. (2019), Enhancement clustering evaluation result of davies-bouldin index with determining initial centroid of k-means algorithm, *in* 'Journal of Physics: Conference Series', Vol. 1235, IOP Publishing, p. 012015.

Kodinariya, T. M., Makwana, P. R. et al. (2013), 'Review on determining number of cluster in k-means clustering', *International Journal* **1**(6), 90–95.

Li, L., Chen, X. and Song, C. (2022), 'A robust clustering method with noise identification based on directed k-nearest neighbor graph', *Neurocomputing* **508**, 19–35.

Liu, G. (2022), 'A new index for clustering evaluation based on density estimation', *arXiv preprint arXiv:2207.01294* .

Madhulatha, T. S. (2012), 'An overview on clustering methods', *arXiv preprint arXiv:1205.1117* .

McNicholas, P. D. (2016), 'Model-based clustering', *Journal of Classification* **33**, 331–373.

Notsu, A. and Eguchi, S. (2016), 'Robust clustering method in the presence of scattered observations', *Neural Computation* **28**(6), 1141–1162.

Rahmah, N. and Sitanggang, I. S. (2016), Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra, *in* 'IOP conference series: earth and environmental science', Vol. 31, IoP Publishing, p. 012012.

Ruspini, E. H. (1969), 'A new approach to clustering', *Information and control* **15**(1), 22–32.

Shah, S. A. and Koltun, V. (2017), 'Robust continuous clustering', *Proceedings of the National Academy of Sciences* **114**(37), 9814–9819.

Sharma, S. (1995), *Applied multivariate techniques*, John Wiley & Sons, Inc.

Tan, P., Steinbach, M. and Kumar, V. (2006), 'Introduction to data mining, addison wesley publishers'.

Tufféry, S. (2011), *Data mining and statistics for decision making*, John Wiley & Sons.

Xu, D. and Tian, Y. (2015), 'A comprehensive survey of clustering algorithms', *Annals of Data Science* **2**, 165–193.

Yang, M.-S., Lai, C.-Y. and Lin, C.-Y. (2012), 'A robust em clustering algorithm for gaussian mixture models', *Pattern Recognition* **45**(11), 3950–3961.

Yuan, C. and Yang, H. (2019), 'Research on k-value selection method of k-means clustering algorithm', *J* **2**(2), 226–235.

Zadeh, L. A. (1965), 'Fuzzy sets', *Information and control* **8**(3), 338–353.

Curriculum Vitae

Eleazer Tabi Serebour was born on February 18, 1996, as the first child of his parent. After completing high school in 2014, he pursued Bachelor of Science in Statistics at Kwame Nkrumah University of Science and Technology (KNUST) in Ghana. He graduated in 2019 with a first-class honor from the Department of Statistics and Actuarial Science. Throughout his time at KNUST, Eleazer held various leadership positions, including serving as the President of Missions Department and Academic Board Chairman of National Union of Presbyterian Students, Ghana. Due to his outstanding academic performance, he received an offer to work as a Teaching and Research Assistant at the Department of Statistics and Actuarial Science, KNUST from September 2019 to August 2020. Eleazer pursued his master's degree in Statistics and Data Science at the University of Texas at El Paso in Fall 2021. He plans to enroll in PhD, which would drive him towards his dream of becoming a top-notch researcher in the field of statistics and other related disciplines. During his studies, Eleazer actively participated in workshops and seminars focused on statistics, data science, and mathematics, which showed his strong interest in these subjects. Additionally, while pursuing his Master's degree, he worked as a Teaching and Research Assistant, making valuable contributions to academic and research activities. After completing his Master's program, Eleazer aims to work in the industry, specializing in data science. His goal is to contribute his skills and knowledge to the field and make an impact in the real-world applications of statistical analysis.

E-mail address: *etabisereb@miners.utep.edu*