Computer Engineering Department
Faculty of Engineering
Deanery of Higher Studies
Islamic University – Gaza
Palestine

Enhancing and Combining a Recent K-means Family of Algorithms for Better Results

Raed Tawfiq Aldahdooh

Supervisor

Dr. Wesam M. Ashour

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1434 هـ - 2013 م

إقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Enhancing and Combining a Recent K-means Family of Algorithms for Better Results

التحسين والجمع بين خوارزميات K-means الحديثة للحصول على نتائج أفضل

أقر بأن ما اشتملت عليه هذه الرسالة إنما هي نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وإن هذه الرسالة ككل، أو أي جزء منها لم يقدم من قبل لنيل درجة أو لقب علمي أو بحثي لدى أية مؤسسة تعليمية أو بحثية أخرى.

DECLARATION

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification

Student's name: **Raed T. Aldahdooh** اسم الطالب: **رائد توفيق الدحدوح**

Signature: التوقيع:

Date: **25/05/2013** التاريخ: **15 رجب 1434 هـ**

بسم الله الرحمن الرحيم

الجامعة الإسلامية – غزة
The Islamic University - Gaza

عمادة الدراسات العليا
هاتف داخلي: 1150

الرقم ج.س.غ/35/..... Ref
التاريخ 2013/05/25..... Date

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ رائد توفيق عادل الدحدوح لنيل درجة الماجستير في كلية *الهندسة* قسم هندسة الحاسوب وموضوعها:

Enhancing and combining a recent K means family of algorithms for Better Results

وبعد المناقشة التي تمت اليوم السبت15 رجب 1434هـ، الموافق 2013/05/25م الساعة الثانية عشرة ظهراً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. وسام محمود عاشور	مشرفاً ورئيساً	د. وسام محمود عاشور .....
أ.د. إبراهيم سليمان أبو هيبة	مناقشاً داخلياً	أ.د. إبراهيم أبو هيبة .....
أ.د. سامي سليم أبو ناصر	مناقشاً خارجيًا	أ.د. سامي أبو ناصر .....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية *الهندسة*/ قسم هندسة الحاسوب.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

عميد الدراسات العليا

أ.د. فؤاد علي العاجز

P.O. Box 108, Rimal, Gaza, Palestine fax: +970 (8) 286 0800 فاكس Tel: +970 (8) 286 0700 ص.ب. 108 الرمال، غزة، فلسطين هاتف
public@iugaza.edu.ps www.iugaza.edu.ps

ACKNOWLEDGMENTS

Firstly, I thank Almighty ALLAH for making this work possible. Then, there are a number of people to whom I am greatly indebted, as without them this thesis might not have been written.

To Dr. Wesam Ashour for his guidance, support, and advice.

To my parents for providing me with the opportunity to be where I am. Without them, none of this would be even possible to do. You have always been around supporting and encouraging me.

I am also very grateful to my friend Walid Alnabahin, who did not spare any effort to review the thesis.

To my brothers, sisters, my wife, my uncle, and friends for their encouragement, input and constructive criticism, which are really priceless.

Table of Contents

List of Figures

List of Tables

التحسين والجمع بين خوارزميات K-means الحديثة للحصول على نتائج أفضل

رائد توفيق الدحدوح

الملخص

تستخدم العنقدة بشكل واسع في العقود الأخيرة لتحليل البيانات وتجميعها، ضمن مجموعات متشابهة في الخصائص، أحد خوارزميات العنقدة هي خوارزمية K-means، التي تعتبر من أكثر خوارزميات العنقدة شيوعاً، ويرجع سبب شهرتها إلى بساطتها من ناحية التطبيق، وسرعتها في ايجاد النتائج، وقدرتها على تحليل وتجميع البيانات وايجاد النتائج الدقيقة.

يقدم الباحث في هذه الأطروحة خوارزمية جديدة أُطلق عليها اسم DIMK-means، وهي عبارة عن خوارزمية تطويرية عن خوارزمية K-means، طور فيها الباحث طريقة جديدة لاختيار المراكز الأولية لبدء عمل الخوارزمية عوضاً عن الطريقة القديمة، وهي طريقة الاختيار العشوائي للمراكز الأولية، والتي كانت تؤول لنتائج سيئة في بعض الأحيان. DIMK-means خوارزمية غير معقدة وتحتاج إلى وقت مكافئ أو أقل من وقت خوارزمية K-means، أضف إلى ذلك أن نتائجها أكثر دقة، و هذا ما يجعلها خوارزمية فعالة من ناحية التطبيق.

بالإضافة إلى ذلك، قدم الباحث في هذه الأطروحة خوارزمية DSMK-means، التي تهدف إلى تجاوز نقاط ضعف خوارزمية K-means عند التعامل مع بيانات تحتوي على مجموعات معقدة في شكلها، ومختلفة في الأحجام والكثافة أو تحتوي على ضوضاء وقيم متطرفة، حيث لجأ الباحث إلى دمج طرق الكثافة في العنقدة وبعض الطرق العملية الأخرى، مثل: الدمج، والتفريق بين المجموعات، للحصول على النتائج الدقيقة في نهاية المطاف. خوارزمية DSMK-means تأخذ وقتاً أكبر في التنفيذ والحصول على نتائج نهائية دقيقة أفضل بكثير من نتائج خوارزمية K-means؛ حيث تتعامل مع بيانات من المستحيل على K-means أن تأتي بنتائج دقيقة عند التعامل معها.

وقد خلصت الدراسة بناءً على مجموعة كبيرة من التجارب إلى أن الخوارزميات التي طورها الباحث قادرة أكثر من غيرها على اكتشاف المجموعات بشكل دقيق من بين مجموعات البيانات المختلفة؛ حتى وإن كانت معقدة في الشكل ومختلفة في الحجم والكثافة أو تحتوي مجموعات غير منفصلة خطيا أو تحتوي على نقاط ضوضاء وقيم متطرفة.

Enhancing and Combining a Recent K-means Family of Algorithms for Better Results

Raed T. Aldahdooh

ABSTRACT

Clustering is widely used to explore and understand large collections of data. K-means clustering method is one of the most popular approaches due to its ease of use and simplicity to implement. In this thesis, the researcher introduces Distance-based Initialization Method for K-means clustering algorithm (DIMK-means) which is developed to select carefully a set of centroids that would get high accuracy results compared to the random selection of standard K-means clustering method in choosing initial centroids, which gets low accuracy results. This initialization method is as fast and as simple as the K-means algorithm itself with almost the same low cost, which makes it attractive in practice.

The researcher also Introduces Density-based Split- and -Merge K-means clustering Algorithm (DSMK-means) which is developed to address stability problems of K-means clustering, and to improve the performance of clustering when dealing with datasets that contain clusters with different complex shapes and noise or outliers.

Based on a set of many experiments, this research concluded that the developed algorithms are more capable to finding high accuracy results compared with other algorithms especially as they can process datasets containing clusters with different shapes, densities, non-linearly separable, or those with outliers and noise. The researcher chose the experiments datasets from artificial and real-world examples off the UCI Machine Learning Repository.

Keywords:

Clustering, K-Means Algorithm, DIMK-means, Cluster Centroid Initialization, Initializing K-Means, K-Means Seeding Technique, DSMK-means, Split and Merge K-means, Density Based K-means, K-means stability, anti-noise K-mean

Chapter 1

1. Introduction

Chapter 1

1. Introduction

This Chapter introduces some necessary background where it identifies simple historical overview of clustering, then explores some important topics, which are: (1) Cluster analysis definition, explanation of its goal and illustration of its difficulty, (2) Definition of terms used throughout the thesis, (3) Researcher motives to carry the research. In addition, the Chapter discusses the thesis contribution.

1.1 Historical Remark

Clustering is a discipline aimed at revealing groups, or clusters, of similar entities in data. The existence of clustering activities can be traced a hundred years back, in different disciplines in different countries. In the mid-18th century, in London during cholera outbreak, John Snow had plotted the diseased reported cases using a special map. A key observation, after the creation of the map, was the close association between the density of disease cases and a single well located at a central street. Without the map; it was very difficult to identify the association between the diseased and their locations. This was the first known application of clustering analysis for many researchers [1].

Since then, cluster analysis consider to be the most popular tool in statistical data analysis which is widely applied in a variety of scientific areas such as data mining, pattern recognition, geographic information systems, information retrieval, microbiology , psychology and other social sciences , in order to identify natural groups in large amounts of data [2] [3].

1.2 What Cluster Analysis Is?

Many definitions of clusters exist. In general terms, Cluster analysis is an important unsupervised learning technique where a set of patterns usually represented as a vector of measurements, or a point in a multidimensional space, is used for identifying groups (clusters) of similar characteristics, Literature review reveals researchers interest in the development of efficient clustering algorithms in a variety of real-life situations,

as indicated by the increase in the number of publications involving this subject in major conferences and journals.

In other terms Morgan Kaufmann define cluster analysis or simply clustering as the process of partitioning a set of data objects or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. The set of clusters resulting from a cluster analysis can be referred to as a clustering. In this context, different clustering methods may generate different clustering's on the same dataset. The partitioning is not performed by humans, but by the clustering algorithm. Hence, clustering is useful in that it can lead to the discovery of previously unknown groups within the data [4]. This definition compared to other definition is a general one while the terms segmentation and partitioning are sometimes used as synonyms for clustering, these terms are frequently used for approaches outside the traditional bounds of cluster analysis. For example, the term partitioning is often used in connection with techniques that divide graphs into sub graphs and that are not strongly connected to clustering. Segmentation often refers to the division of data into groups using simple techniques; e.g,. an image can be split into segments based only on pixel intensity and color, or people can be divided into groups based on their income. Nonetheless, some work in graph partitioning and in image and market segmentation is related to cluster analysis [5].

The term "clustering" is most popular and used in several research communities to describe methods for grouping of unlabeled data.

1.3 Definitions

The following terms are used throughout the thesis:

- **Dataset:** A dataset is a collection of data, usually presented in tabular form. Each column represents a particular (variable, or attribute). Each row corresponds to a given member of the dataset called (object, or point).
- **A Cluster**: is a well-defined collection of objects, which are "similar" among itself and are "dissimilar" to the objects from other clusters.
- **Well-Separated Clusters:** Clusters where each object in a cluster is closer "similar" to every other object in the same cluster than to any object in other clusters. Figure 1.1 show three well-separated clusters of two-dimensional objects.

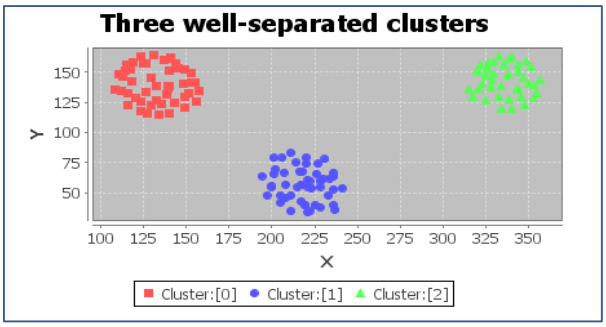

Figure 1.1 : Three well-separated clusters of two-dimensional objects.

- **Centroid or prototype:** A objects in a cluster is called a centroid when it is located in the center of the cluster; this point can be identified as the average of all the objects in the cluster, or the Medoid, which is the most "representative" point of the cluster.
- **Prototype-Based "Center-based" Clusters:** A cluster is center-based when its objects are closer "more similar" to its "center", than to the center of any other cluster. Figure 1.2 show four centroid-based clusters of two-dimensional objects.

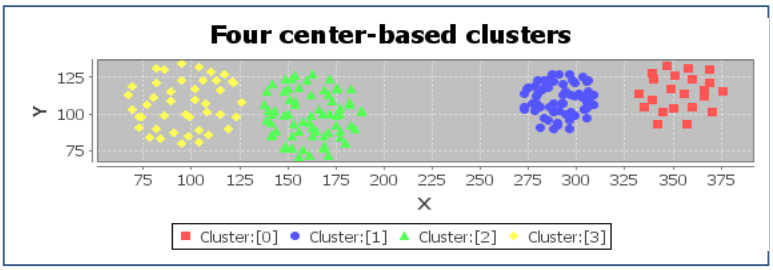

Figure 1.2: Four center-based clusters of two-dimensional objects.

- **Contiguous Cluster (Nearest neighbor or Transitive Clustering):** The cluster where its object is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster. Contiguous clusters of two-dimensional objects are shown in Figure 1.3.

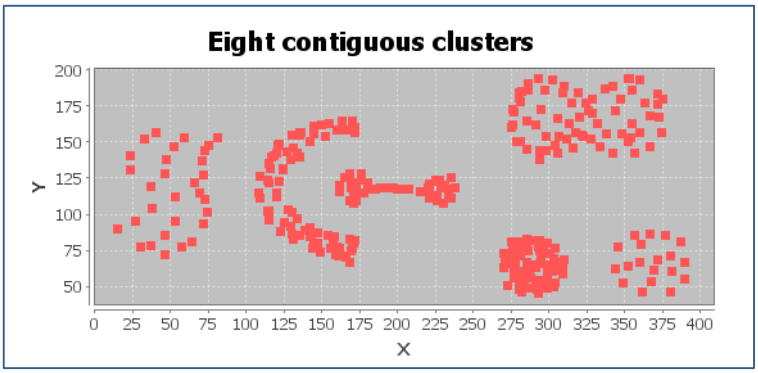

Figure 1.3: Eight contiguous clusters of two-dimensional points.

- **Density-based clusters:** A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density. Density-based clusters of two-dimensional objects are shown in Figure 1.4.

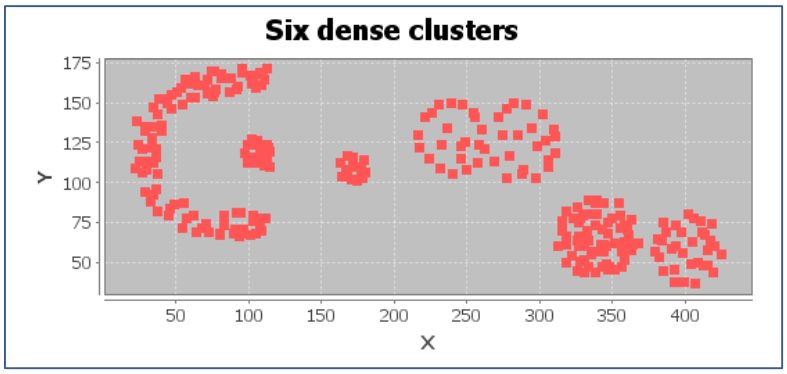

Figure 1.4: Six clusters of two-dimensional points.

- **Exclusive (hard or crisp) clustering:** each data object can only exist in one cluster.
- **Overlapping:** allows data objects to be grouped in 2 or more clusters.
- **A Fuzzy clustering:** assigns each object to each cluster with a certain degree of membership.
- **Complete clustering:** assigns every object to a cluster.
- **Partial clustering:** allows some data objects to be left alone.
- **Cluster Seed:** First centroid of a cluster which is defined as the initiator of that cluster.
- **Outlier / Noise:** We can identify Outlier as a noisy observation (objects or points), which does not fit to the assumed model that generated the data. Alternatively, in other definition, outliers are considered as observations that should be removed in order to make clustering more reliable.
- **Noisy Dataset**: is a dataset whose data records inaccurately represent some is meaningless data records.
- **Density area or unit**: an area is considered dense or not based on a value defined by the number of its neighbor points "MinPts" within a given radius "ε". Such radius and MinPts are calculated dynamically.
- **A distance measure (a specialization of a proximity measure):** is a metric (or quasi-metric) on the feature space used to quantify the similarity of patterns.

- **The complete linkage clustering (or the farthest neighbor method):** is a method of calculating and finding maximum distance between a pair of objects one in one cluster, and one in the other.
- **The single linkage clustering (nearest neighbor or shortest distance):** is a method of calculating and finding minimum distance between a pair of objects one in one cluster, and one in the other "closest objects".
- **Data Types and Scales:** The attributes of the objects can be of different data types and can be measured on different data scales. Data scales and types are important since the type of clustering used often depends on the data scale and type.
    - **The different types of attributes are**
        i. Binary (two values)
        ii. Discrete (a finite number of values)
        iii. Continuous (an effectively infinite number of values)
    - **The different data scales are**
        i. Qualitative
            (1) Nominal – the values are just different names.
            (2) Ordinal – the values reflect an ordering, nothing more.
        ii. Quantitative
            (3) Interval – the difference between values is meaningful, i.e., a unit of measurement exits.
            (4) Ratio – the scale has an absolute zero so that ratios are meaningful.

1.4 Research Questions

The research questions addressed in this thesis include:

1. Could we improve recent K-means analysis by making it less sensitive to noise, cluster shape, and data size?
2. How can we improve recent K-means algorithms initialization process, which has big impact on algorithms results?
3. Does the initialization process need to use parameters? Can we make algorithms determine the parameters depending on the nature of data?

1.5 Motivation

Data analysis underlies many computing applications, either in a design phase or as part of their on-line operations. Data analysis procedures can be dichotomized as either exploratory or confirmatory, based on the availability of appropriate models for the data source, but a key element in both types of procedures (whether for hypothesis formation or decision-making) is the grouping, or classification of measurements based on either (i) goodness-of-fit to a postulated model, or (ii) Natural groupings (clustering) revealed through analysis [6].

Clustering is useful in several exploratory pattern-analysis, grouping, decision- making, and machine-learning situations; including data mining, document retrieval, image segmentation, and pattern classification. It is the process of producing unlabeled categorized data. However, On trajectory data clustering is a very important data mining task for a wide variety of application fields including location aware services, geo-marketing protein analysis etc. Most of traffic planner or Geo-marketer takes interest to know the most visited place or important place with respect to product promotion; based on this, clustering is very useful in various applications [7].

K-means is one of the most famous partition clustering algorithms because of: (i) It has been recently elected and listed among the top ten most influential data mining algorithms; (ii) it is at the same time very simple and slightly scalable, as it has linear asymptotic running time with respect to any variable of the problem. K-means clustering is a method of cluster analysis, which aims to partition n observations ($x_1, x_2 \ldots x_n$), where each observation is a d-dimensional real vector into k clusters in which each observation belongs to the cluster with the nearest mean. In general, K-means is one of the most important and best performances of the clustering algorithms. However, there are some drawbacks for K-means algorithm like sensitivity to the initial cluster centroids, which is addressed in these references [8] [9]. Moreover, when the number of data points is large, it takes enormous time to find the global optimal solution [10].

K-means has several limitations which are listed below :

- **Scalability:** It scales poorly computationally.
- **Initial Centroids:** The clustering result is extremely sensitive to the initial centroids.
- **Noise**: Noise or outliers deteriorates the quality of the clustering result.

- **Number of clusters**: The number of clusters must be determined before the means clustering begins.
- **Local minima**: It always converges to different local minima based on the initializations process.
- **Inability to cluster non-linearly separable dataset**: It fails to split non-linearly separable datasets in the input space.

It is very convenient to classify algorithms based on the relative amount of time or relative amount of space they require and specify the growth of time/space requirements as a function of the input size. Thus, The K-means time complexity is O(NKI), (*where N is number of objects, K is the number of clusters , and I is the number of iteration taken by the algorithm until convergence criterion is satisfied*) And its space complexity is O(K+N), as it requires additional space to store the data matrix. Add to that K-means order-independent; for a given initial seed set of cluster centers, it generates the same partition of the data irrespective of order in which pattern are presented to the algorithm. Because of these characteristics, K-means algorithm is considered as one of the top ten most influential data mining algorithms, which is one reason that encouraged the researcher to choose K-means clustering to be the focus of this thesis.

There are a large number of researchers up to this moment try to develop and enhance K-means algorithm to optimize the performance and overcome algorithm drawbacks. These reasons and others prompted the researcher to choose this algorithm. However, the researcher aims at improving the performance of this algorithm by creating new initialization process "seeding process", which will contribute to overcome the initial centroid sensitivity drawback. Moreover, the researcher will combine some of the recent K-means family of algorithms to optimize the algorithm results and preserve its stability in addition to reduce its sensitivity towards noise.

1.6 Thesis Contribution

This thesis contributes to the area of pure experimental computer science; specifically, it introduces novel thinking and techniques to the fields of partition based clustering techniques. The primary objective of this thesis is to optimize the performance of K-means clustering algorithm, which is considered as one of the top ten partition based clustering algorithms in data mining.

The contribution of this thesis is two-fold:

(1) The researcher will develop a new clustering algorithm called DIMK-means "Distance-based initialization method for K-means clustering algorithm" that will be developed to address sensitivity of the algorithm for selecting initial means "careful seeding", and reducing the effect of outliers or noise.

(2) Also the researcher will develop another clustering algorithm called DSMK-means "Density-based Split-and-Merge K-means clustering Algorithm" that will be developed to address stability of K-means clustering problems, and to decrease deterioration of the clustering results quality because of cluster shape, size, noise or outliers.

The researcher will evaluate the proposed algorithms using real and artificial data and compare algorithms' results with other famous related algorithms' results. It is expected that the results of the proposed algorithms will confirm the high performance of the proposed methods in both quality and time.

1.7 Organization of The Thesis

This thesis has five chapters, which will give an overview of the thesis where relevant, the researcher highlights the major issues addressed in the chapters, and what researcher regards as the key contributions of the work; the following is a brief description of the content of each chapter:

Following on from this introduction, **Chapter 2** reviews the related work, which discusses the clustering problems and background. The Chapter includes also a description of standard K-means algorithm, highlights the effects of random selection for initial cluster centroids, and the effects of different cluster shapes and noise or outlier on the quality of algorithm results.

Chapter 3 overviews and discusses the new proposed algorithms. Furthermore, it describes some of the most important terms related to the proposed algorithms. A summary of the research methodology and design are provided in this chapter.

Chapter 4 illustrates the experiment, its results and the analysis of these results. In addition, the Chapter explains the means of measuring the algorithm results' quality and presents a comparison with other algorithms' results.

Chapter 5 This final Chapter discusses a general summary and offers conclusions of the thesis in addition to proposed future

Chapter 2

2. Literature Review

Chapter 2

2. Literature Review

This Chapter presents necessary background and related work. First, it identifies simple background overview. Then explores some important topics: (1) requirements for cluster analysis (scalability, discovery of clusters with arbitrary shape, ability to deal with noisy data, etc…), (2) defines a categorization of major clustering methods, (3) K-means algorithm, finally, it discusses related work, which is divided into two main research [i]. Research that addresses K-means initialization methods and [ii]. Research that addresses K-means stability in results and sensitivity to outliers

2.1 Background

The main goal of clustering is to reduce the amount of data by categorizing or grouping similar data items based on an underlying measure of similarity. Such grouping is pervasive in the way human's process information, and one of the motivations for using clustering algorithms is to provide automated tools to help in constructing categories or taxonomies. These methods may also be used to minimize the effects of human factors in the process [11]. The cluster Analysis has been used for the following three main purposes [12].

- **Underlying structure:** to gain insight into data, generate hypotheses, detect anomalies, and identify salient features.
- **Natural classification:** to identify the degree of similarity among forms or organisms (phylogenetic relationship).
- **Compression:** as a method for organizing the data and summarizing it through cluster prototypes.

Clustering is a difficult problem and in order to better understand the difficulty of deciding what constitutes a cluster, consider Figures 2.1.(a) through 2.1.(d), which show twenty points and three different ways in which these points can be divided into clusters. If we allow clusters to be nested, then the most reasonable interpretation of the structure

of these points is that they can be divided into two clusters, each of which has three sub clusters. However, the apparent division of the two larger clusters into three sub clusters may simply be an artifact of the human visual system, but it may be reasonable also to say that the points form four clusters. Thus, we stress once again that the definition of what constitutes a cluster is imprecise, and the best definition depends on the type of data and the desired results [13].

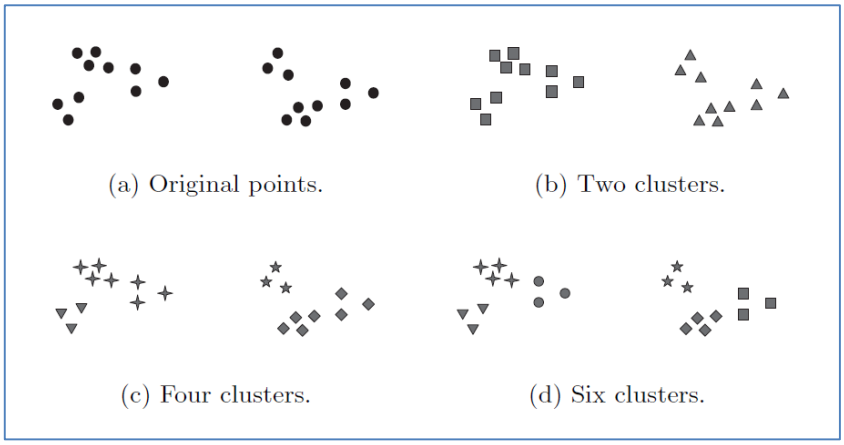

Figure 2.1: Different ways to clustering the same set of points [13].

2.1.1 Requirements for Cluster Analysis

Clustering is a challenging field of research in which its potential applications pose their own special requirements. Although many researchers defined the requirements for clustering, Han & Kamber have the most suitable definition of the requirements that are listed below [4]:

1. **Scalability:** Many clustering algorithms work well on small datasets containing fewer than 200 data objects. However, a large database may contain millions of objects. Clustering on a sample of a given large dataset may lead to biased results. Highly scalable clustering algorithms are needed.
2. **Ability to deal with different types of attributed:** Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.
3. **Discovery of clusters with arbitrary shape:** Many clustering algorithms determined clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures 'end to find spherical clusters with similar size and density'. However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape.

4. **Minimal requirements for domain knowledge of determine input parameters:** Many clustering algorithms require users to input certain parameters in cluster analysis (such as the number of desired clusters). The clustering results can be quite sensitive to input parameters. Parameters are often hard to determine, especially for datasets containing high-dimensional objects. This not only burdens users, but also makes the quality of clustering difficult to control.
5. **Ability to deal with noisy data:** Most real-world databases contain outliners or missing, unknown, erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.
6. **Insensitivity to the order of input records:** Some clustering algorithms are sensitive to the order of input data; for example, may generated dramatically different clusters. It is important to develop algorithms that are insensitive to the order of input.
7. **High dimensionality:** A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions. Human eyes are good at judging the quality of clustering for up to three dimensions. It is challenging to cluster data objects in high-dimensional space, especially considering that such data can be very sparse and highly skewed.
8. **Constraint-based clustering:** Real-world applications may need to perform clustering under various kinds of constraints. Suppose that your job is to choose the locations for a given number of new automatic cash-dispensing machines (ATMs) in a city. To decide upon this, we may cluster household while considering constraints such as the city's rivers and highway networks and customer requirements per region. A challenging task is to find groups of data with good clustering behavior that satisfy specified constraints.
9. **Interpretability and usability:** Users expect clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied up with specific semantic interpretations and applications. It is important to study how an applications goal may influence the selection of clustering methods.

2.1.2 A Categorization of Major Clustering Methods

To satisfy the requirements of clustering; different clustering methods have been developed, each of which uses a different induction principles, and gives different grouping of a dataset. Deciding which the most suitable method is depends on the type of the output desired, the known performance of a certain method with particular types of data, the hardware and software facilities available, and the size of the dataset. In general; clustering methods have different categorization, Farley and Raftery (1998) suggest dividing the clustering methods into two main groups: hierarchical and partitioning methods. Han and Kamber (2001) suggest categorizing the methods into additional three main categories: density-based clustering, model-based clustering and grid-based clustering. An alternative categorization method based on the induction principles of the various clustering methods is presented in (Estivill-Castro, 2000) [14]. Several studies examine a lot of clustering techniques, of which the researcher found most efficient categorization techniques are those organized into the following categories: partitioning, hierarchical, grid-based, density-based, model-based, methods for high-dimensional data, and constraint-based clustering techniques.

- **Partition-based clustering** attempts to directly decompose the dataset into a set of disjoint clusters. The criterion function that the clustering algorithm tries to minimize may emphasize the local structure of the data, as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters. Cluster similarity is measured in regard to the mean value of the objects in a cluster, center of gravity, (K-means [15]) or each cluster is represented by one of the cluster objects located near its center (K-Medoid [16]). The most popular and the simplest partitional algorithm is K-means. Since partitional algorithms are preferred in pattern recognition due to the nature of available data, our coverage here is focused on these algorithms. K-means has a rich and diverse history as it was independently discovered in different scientific fields. Even though K-means was first proposed over 50 years ago, it is still one of the most widely used algorithms for clustering. Ease of implementation, simplicity, efficiency, and empirical success are the main reasons for its popularity [12].

❖ **Hierarchical clustering algorithms** recursively find nested clusters either in agglomerative mode (starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy) or in divisive (top-down) mode (starting with all the data points in one cluster and recursively dividing each cluster into smaller clusters), agglomerative mode vs. divisive mode are described in Figure 2.2 [12]. In other terms, hierarchical clustering proceeds successively either by merging smaller clusters into larger ones or by splitting larger clusters. The clustering methods differ in the rule by which it decides which two small clusters are merged or which large clusters are split. The end result of the algorithm is a tree of clusters called a dendrogram, which shows how the clusters are related. By cutting the dendrogram at a desired level; a clustering of data items into disjoint groups is obtained; Hierarchical algorithms such as BIRCH [17] and CURE [18].

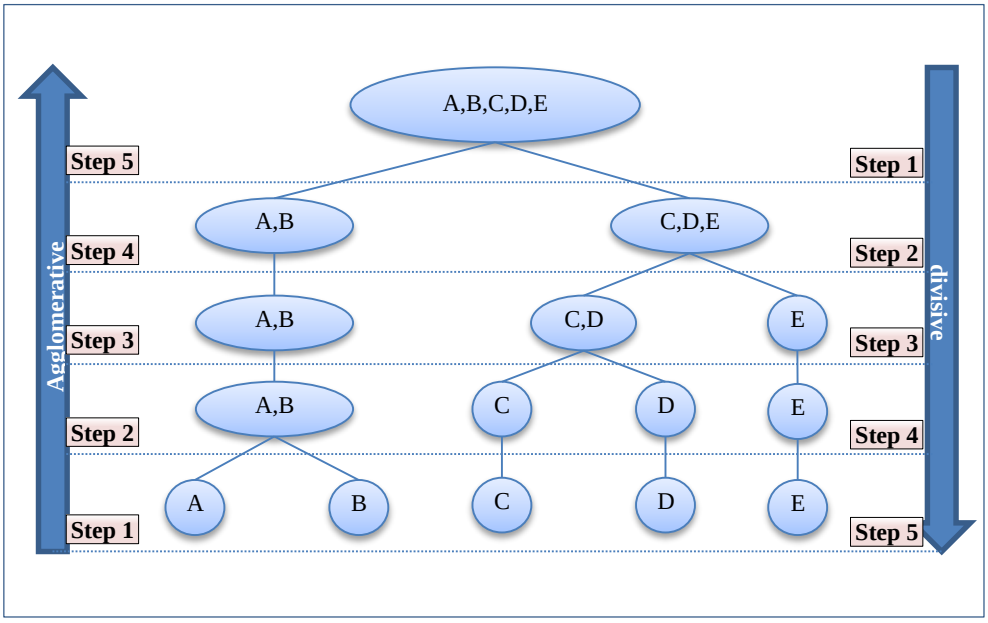

Figure 2.2: Agglomerative and divisive clustering

❖ **Grid-based clustering** methods make it possible to form arbitrarily shaped, distance independent clusters. In these methods, the feature space is quantized into cells using a grid structure. The cells can be merged together to form clusters. Grid-based clustering was originally based on the idea of Warnekar and Krishna to organize the feature space containing patterns [19]. Schikuta has used topological neighbor search algorithm to combine the grid cells to form clusters [20]. CLIQUE [21], named for Clustering In Quest, is a density and grid-based approach for high dimensional datasets that provides, automatic sub-space clustering of high dimensional data. Grid-based algorithms such as STING [22], and WaveCluster [23], are based on multi-level grid structure on which all clustering operations are performed.

- **In Density-based clustering** [24], clusters are defined as areas of higher density than the remainder of the dataset. The most popular density based clustering method is DBSCAN [25]. In contrast to many newer methods, it features a well-defined cluster model called "density-reachability". Similar to linkage based clustering; it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range. OPTICS [26] is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter a, and produces a hierarchical result related to that of linkage clustering.
- **A model-based method** hypothesizes a model for each of the clusters and finds the best fit of the data to that model. Examples of model-based clustering include the EM algorithm (which uses a mixture density model), conceptual clustering (such as COBWEB [27]) and neural network approaches (such as self-organizing feature maps).
- **Clustering high-dimensional data** is of crucial importance, because in many advanced applications; data objects such as text documents and microarray data are high-dimensional in nature. There are three typical methods to handle high dimensional datasets: dimension-growth subspace clustering represented by CLIQUE [20], dimension reduction projected clustering, represented by PROCLUS, and frequent pattern–based clustering, represented by pCluster.
- **A constraint-based clustering method** groups objects based on application dependent or user-specified constraints. For example, clustering with the existence of obstacle objects and clustering under user-specified constraints are typical methods of constraint-based clustering.

2.1.3 Partition Based Clustering:

A partition clustering algorithm splits the data points into k partitions, where each partition represents a cluster. The partitioning is done based on certain objective function. One of the criterion functions is minimizing square error criterion, which is computed as shown by equation 2.1:

$$\mathbf{E} = \sum_{K=1}^{K} \sum_{\forall P_i \in C_K} \left|\left|P_i - \mu_K\right|\right|^2 \qquad (2.1)$$

Where C_K is the set of instances in cluster K; μ_K is the prototype of cluster k.

Each K cluster must have at least one point and each point must be in one and only one cluster.

2.1.4 K-Means Algorithm

K-means is one of the most widely used partition-based clustering algorithms in practice. It is simple, easy, understandable, scalable, and can be adapted to deal with streaming data and very large datasets [28]. K-means algorithm divides a dataset X into K disjoint clusters based on the dissimilarities between data objects and cluster centroids. Let $\bar{\mu}_i$ be the centroid of cluster Ci and the distances between Xj that belong to Ci and $\bar{\mu}_i$ is equal to d(Xj, $\bar{\mu}_i$). Then, the objective function minimized by K-means is given by:

$$\min_{\bar{\mu}_1,..,\bar{\mu}_k} \mathbf{E} = \sum_{i=1}^{K} \sum_{x_j \in C_i} \mathbf{d}\left(\mathbf{x_j}, \bar{\mu}_i\right) \qquad (2.2)$$

Where 'd' is one of distance function. Typically d is chosen as the Euclidean or Manhattan distance.

The Euclidean distance between points X and Y is the length of the line segment connecting them ($\overline{XY}$). If X and Y are n-dimensional vectors where X= (x_1, x_2,..., x_n) and Y = (y_1, y_2,..., y_n), then the Euclidean distance from X to Y, or from Y to X is given by:

$$\begin{Bmatrix} d(X,Y) \\ d(Y,X) \end{Bmatrix} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (2.3)$$

The Manhattan distance between two points measured along axes at right angles where distance that would be traveled to get from one data point to the other if a grid-like path is followed. In a plane with X at (x_1, x_2) and Y at (y_2, y_2), it is $|x_1 - y_1| + | x_2 - y_2|$. The Manhattan distance between two n-dimensional vectors is the sum of the differences of their corresponding components.

$$d(X,Y) = \sum_{i=1}^{n} |x_i - y_i| \qquad (2.4)$$

Where n is the number of variables, and Xi and Yi are the values of the ith variable, at points X and Y respectively.

Usually the selection process between the two methods of calculating the distance is left to the user based on the nature of the data. Figure 7 shows the difference between using Euclidean and Manhattan distance to calculating the distance between two points in two-dimensional space.

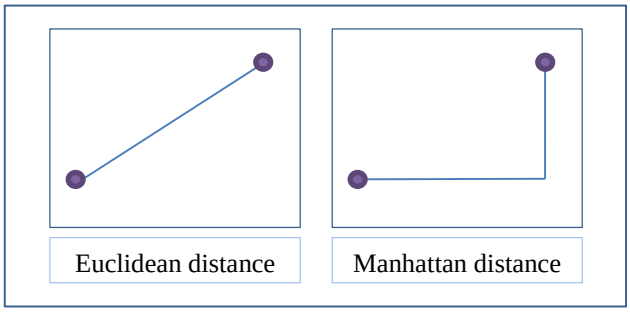

Figure 2.3: Euclidean and Manhattan distance between two point in tow-dimensional space.

K-means algorithm working process summarized as follows:

1. Determine the number of clusters (k parameters in k-means).
2. K-means selects randomly k cluster centroids.
3. Assign object to clusters based on distance function.
4. When all objects have been assigned, Re-compute new cluster centroids by averaging the observations assigned to a cluster.
5. Repeat (3-4) until convergence criterion is satisfied.

Pseudo code for K-means algorithm:

Algorithm 2.1: K-means

Input: $X=\{x_1,x_2,\ldots,x_j\}$ (set of entities to be clustered)

K (number of cluster)

MaxIters (Limit of iterations)

Output: C= $\{\overline{\mu_1}, \overline{\mu_2}, .., \overline{\mu_k}\}$ (set of cluster centroids)

L= (set of cluster labels of X)

1. *Require:* $k \geq 2$ *and* $t \geq 1$ $\begin{cases} k: number\ of\ cluster, \\ t: max\ number\ of\ iteration. \end{cases}$
2. *Select initial cluster centroids* $\overline{\mu_1}, \overline{\mu_2}, .., \overline{\mu_k}$.
3. *Repeat*
4. *For each point* x_j *in a dataset do*
5. *For all* $\overline{\mu_\iota}$ *do*
6. *Compute the dissimilarity* $d(x_j, \overline{\mu_\iota})$;
7. *End for.*
8. *assign point* x_j *to closest cluster* C_i;
9. *End for.*
10. *For all* $\overline{\mu_\iota}$ *do*
11. *Update* $\overline{\mu_\iota}$ *as the centroid of cluster Ci;*
12. *End for.*
13. *Until convergence criterion is satisfied or the number of iterations exceeds a given limit t.*

The number of clusters found is equal to the number of the initial starting points, which are specified as input parameters to the clustering algorithm.

2.2 Related Work

K-means clustering algorithm has a very rich history because of its observed speed and simplicity, in this work the focus is on improving its accuracy. In the following sub sections, the researchers reviews initialization methods in K-means algorithm in addition to different research studies that were developed to enhance K-means accuracy.

2.2.1 K-Means Initialization Methods

The initial location of the cluster centroid has major impact on the performance of K-means algorithm. These effects will be discussed in the following sub section. The

following are some methods proposed by different researchers that decrease the sensitivity and increase accuracy of K-means, through selection of the best centroid locations within the existing dataset.

❖ **K-means++: the advantages of careful seeding** [29]**:**

In 2007; David Arthur and Sergei published research titled "K-means++: The Advantages of Careful Seeding" where they proposed a specific way of choosing initial centroids. In their proposed method, initial centroids are chosen consecutively with probability proportional to the distance to the nearest centroid as follows:

1. Choose an initial centroid c1 =x randomly from X.
2. Set D(x) as the shortest Euclidean distance from a data point x to the closest centroid.
3. Choose the next centroid ci, selecting ci = $x' \epsilon X$ with probability $\frac{D(x')^2}{\sum D(x)^2}$
4. Repeat steps 2 and 3 until we have chosen a total of K centroids.
5. Proceed as with the standard K-means algorithm.

This seeding method yields considerable improvement in the final error of k-means. Although the initial selection in the algorithm takes extra time, The authors tested their method with real and synthetic datasets and obtained typically 2-fold improvements in speed, and for certain datasets, close to 1000-fold improvements in error. In these simulations, the new method usually performed at least as well as standards k-means in both speed and error. In summary K-means++ presented a new way to seed the K-means algorithm that is O(logk) competitive with the optimal clustering. Where the initialization needs k passes over the data, which make it does not scale very well to large data sets.

❖ **Initializing partition-optimization algorithms** [30]:

Initializing Partition-Optimization Algorithms proposes a staged approach to specifying initial values by finding a large number of local modes and then obtaining representatives from the most separated ones. The researcher propose Amulti-Stage Initializer; the steps of algorithm are outline below:

Let X be the n × p data matrix with rows given by the observations $X = \{X_1, X_2, ..., X_n\}$. the algorithm objective is to find initial seeds for partitioning algorithms to group the dataset into K clusters, assuming that K is known. Consider the following multi-stage algorithm:

1. Obtain the singular value decomposition (SVD) of the centered data $X^{*}=U\,DV^{'}$, where D is the diagonal matrix of the m positive singular values $d1 \geq d2 \geq ... \geq d_m$, and U and V matrices of order $n \times m$ and $p \times m$, both with orthonormal columns (in n-and p dimensional space, respectively). For a given m^{*} consider the reduced $n \times m^{*}$ projection given by U^{*} consisting of the first m^{*} columns of U given by $u_1, u_2, \ldots, u_{m^*}$. We propose working in the reduced space.
2. For each coordinate in the reduced space, we obtain an appropriate number of local modes. We choose more modes in those coordinates with higher singular values (or standard deviations of the principal components), under the assumption that information in the dataset is more concentrated along those projections corresponding to higher values, and therefore these would contain more information about the clusters. Specifically, we propose choosing the number of modes, k_j in the *j*th reduced-space coordinate to be equal to $\left\lceil (c_{m-m*}\,K)^{1/m*} \right\rceil d_j/d_{m*}$ rounded to the nearest integer, with $\lceil x \rceil$denoting the smallest integer greater than or equal to x, and c_k is non-decreasing and concave in k. They propose one-dimensional k-means to determine the modes in the *j*th reduced coordinate data space initialized using the quintiles corresponding to the k_j equal increments in probabilities in (0,1). The choice of k-means is appropriate because the goal here is to find a large number of univariate local modes for input into the next step.
3. Form the set of candidate multivariate local modes in the reduced space by taking the product set of all the one-dimensional modes. Eliminate all those candidates from the product set which are not closest to any observation in $U*$. The remaining k^{*} modes are used as initial points for a K-means algorithm that provides us with k^{*} local modes. Note that typically, $k^{*} >> k$.
4. Obtain the k^{*} local modes of the dataset using the K-means algorithm with the starting points provided from above. Also, classify the observations, and obtain the corresponding group means in the original domain.
5. At this point, we have k^{*} local modes of the dataset in the reduced space and the corresponding group centers in the original space. The goal is to obtain k representative points from the above, which are as far as possible from each other. We use hierarchical clustering with single-linkage on these k^{*} modes and cut the tree into k groups. Since a single-linkage merge criterion combines

groups based on the least minimum pairwise distance between its members, its choice in the hierarchical clustering algorithm here means that we obtain k groups of local modes (from out of k^*) that are as far apart in the transformed space as possible. Means, and if needed, relative frequencies and dispersions, of the observations in the dataset assigned to each of the k grouped modes are calculated: these provide the necessary initialization points for the partition-optimization algorithms.

The main contribution of this research is the development of a computationally feasible deterministic algorithm for initializing greedy partition-optimization algorithms. The results on an extensive suite of test experiments and a classification dataset are very promising. Where, the computation complexity is high.

- **Cluster Center Initialization Method for K -means Algorithm Over Datasets with Two Clusters** [31]:

Cluster Center Initialization Method for K-means Algorithm Over Datasets with Two Clusters defines nearest neighbor pair and puts forward four assumptions about nearest neighbor pairs, based on which a centroid initialization method for K-means algorithm over datasets with two clusters is build. The steps of research are outlined below:

Supposing that $X=\{x_1,x_2,...,x_n\}$ is a dataset, where $xj=\{x_{1j},x_{2j},...,x_{mj}\}T$.

1. Compute the dissimilarity between any pair of data points in X using formula:

$$d(x_j, x_k) = \sqrt{(x_j - x_k)^T (x_j - x_k)}$$

2. For any datum point x in X find its nearest neighbor x_{NN} using formulae: $x_{NN} = \arg\min_{y \epsilon X - \{x\}} \{d(x, y)\}$ and constitute a set B of nearest neighbor pairs

3. Find two most dissimilar nearest neighbor pairs, (x_1, $x_{1,NN}$) and (x_2, $x_{2,NN}$), using formulas:

$$d = \left((x, x_{NN}), (y, y_{NN})\right)$$

$$d = \min\left\{d(a, b) \middle| a \in \{x, x_{NN}\}, b \in \{y, y_{NN}\}\right\}$$

$$d' = \left((x_1, x_{1,NN}), (x_2, x_{2,NN})\right)$$

$$d' = \max\left\{d\left((x, x_{NN}), (y, y_{NN})\right) \middle| (x, x_{NN}) \in B, (y, y_{NN}) \in B\right\}$$

4. Find the third most dissimilar nearest neighbor pairs (x_3, $x_{3,NN}$).
5. Find the fourth most dissimilar nearest neighbor pairs (x_4, $x_{4,NN}$).
6. Find the nearest neighbor pair (x_5, $x_{5,NN}$) on the overlapping of two clusters..
7. Select two initial cluster centroids according to some assumptions.

Cluster center initialization method CIT devotes to searching two nearest neighbor pairs that are most dissimilar and in different clusters, but not on the overlapping of two clusters. The means of each searched nearest neighbor pairs are selected as two initial cluster centers.

❖ **Hierarchical K-means: an algorithm for centroids initialization of K-means** [32]:

Hierarchical K-means: an algorithm for centroids initialization for K-means, a new approach to optimize the initial centroids for K-means proposed. It utilizes all the clustering results of K-means in certain times, even though some of them reach the local optima. Then, transform the result by combining with Hierarchical algorithm in order to determine the initial centroids for K-means. The execution steps of the proposed Hierarchical K-means algorithm to determine initial centroids for K-means are described as follows:

1. Set $X=\{x_i \mid i=1,\ldots,r\}$ as each data of A, where $A=\{a_i \mid, i=1,...,n\}$ is attribute of n-dimensional vector.
2. Set K as the predefined number of clusters.
3. Determine p as numbers of computation
4. Set i=1 as initial counter
5. Apply K-means algorithm.
6. Record the centroids of clustering results as $Ci= \{cij \mid j=1,\ldots,K\}$
7. Increment i=i+1
8. Repeat from step 5 while i<p.
9. Assume $C = \{ Ci \mid i=1,\ldots,p\}$ as new dataset, with K as predefined number of clusters
10. Apply hierarchical algorithm

11. Record the centroids of clustering result as D = {di | i=1,…,K}

Then, apply D = {di | i=1… K} as initial cluster centroids for K-means clustering. The experiment results reflect the accuracy of the method.

❖ **Efficiency issues of evolutionaryK-means** [33]:

Efficiency issues of evolutionary K-means method suggest that evolutionary techniques conceived to guide the application of K-means can be more computationally efficient than systematic (i.e., repetitive) approaches that try to get around the K-means drawbacks by repeatedly running the algorithm from different configurations for the number of clusters and initial positions of prototypes. To do so, a modified version of a (K-means-based) fast evolutionary algorithm for clustering is employed. From the theoretical perspective, the time complexity of all the assessed algorithms has been demonstrated to be linear with respect to the number of data objects and attributes. This method suggests that, in principle, all of them are eligible to be employed in real world applications involving large datasets. Furthermore, this method has shown that well-designed evolutionary algorithms for clustering are also promising tools for real-world practical applications in which computational efficiency is of paramount importance.

❖ **A Deterministic Method for Initializing K-means Clustering** [34]:

A Deterministic Method for Initializing K-means Clustering by Ting Su and Jennifer Dy motivate theoretically and experimentally the use of a deterministic divisive hierarchical method, which they refer to as PCA-Part (Principal Components Analysis Partitioning) for initialization. The researchers proposed sorting data instances on a single variable then performed the initial partition. These partitions are used only in one dimension. An alternative method is to partition the sample space hierarchically. Starting with one cluster, then cut it into two. Pick the next cluster to partition, and so on. PCA-Part uses the latter approach. The performance of K-means depends on the initial condition. According to researchers, results are encouraging. It presents some promise in initializing at intelligent starting points for the K-means algorithm, instead of just random start.

❖ **Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering** [35]**:**

A Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering by Renato Cordeiro de Amorim and Boris Mirkin represents another step in overcoming a drawback of K-Means, its lack of defense against noisy features, using feature weights in the criterion. In this criterion, two modifications of weighted method are proposed and their competitiveness is experimentally demonstrated. The main contribution of this research is the extension of the exponent β from the weights in the original Weighted K-means method to the distances, in the form of Minkowski metric criterion. This returns the K-means criterion to its original format of summary distances between entities and their cluster centroids and makes the weights to be the feature rescaling coefficients. The Minkowski metric criterion does the job: in the experiments, it consistently improves the accuracy of the Weighted K-means both at the original and noisy datasets. The issue remaining to be addressed in this regard, as it is with the original Weighted K-Means, is of determining the right value of β exponent. Applying a semi-supervised setting by training β on labeled subsamples appears to be a promising direction. Another possibility would lie in trying to identify characteristics of the data structures that relate to specific values of β. Furthermore, a related contribution of this paper is the usage of anomalous cluster centers to initialize both centroids and feature weights in the ‘‘intelligent’’ versions of the Weighted K-Means. This proved effective at modest to moderate data sizes.

By the end of this subsection, the researcher criticizes most of the researches mentioned above, where many of these algorithms proposed to solve the sensitivity of K-means to centroids initialization process that has a direct impact on the formation of final clusters. Most of the algorithms mentioned above suffer from high computational complexity; therefore, they do not have strong scalability. This has led the researcher attempt to develop a new simple and scalable algorithm to decrease the sensitivity of centroids initialization process**.**

2.2.2 K-Means Stability in Results and Sensitivity to Outliers

This sub section is mainly concerned with presenting the algorithms that enhance and improve the performance of K-means. We will review methods that decrease the sensitivity of algorithm towards outlier or noise, and other related methods.

❖ **BNAK-Divide-and-Merge Clustering Algorithm** [36]:

Divide-and-Merge is a methodology for clustering a set of objects that combines a top-down "divide" method with a bottom-up "merge" method. This algorithm proposes a normalized cut with automatically determining K clustering algorithm (BNAK-Divide-and-Merge) based on the Divide-and-Merge. Like the Divide-and-Merge, there are also two phases in this approach.

i. Divide phase:

Which is the first phase of Divide-and-Merge Algorithm, applies the spectral clustering algorithm to form a tree T whose leaves are the objects. A new threshold is proposed and called minDividedSize in Step 1 to control the number of tree nodes produced by the divide phase, which can greatly improve the efficiency of the divide phase. In Step 2, D is the diagonal matrix of the row sums of similarity matrix $A \bullet A^T$.

Pseudo code for dividing phase:

Input: An $m\ n\times$ matrix A and a threshold minDividedSize

Output: A tree whose leaves are subsets of the objects

1. *If the size of A is not less than minDividedSize, then go to step 2, else stop.*
2. *Compute the Laplacian matrix $L=D-A \bullet A^T$.*
3. *Compute the two smallest eigenvectors V_1 and V_1 of $D^{-1}L$, let $V=\{y_1,y_2,..,y_n\}^T$ where $V=\{v_1,v_2\}$*
4. *Partition the samples $y_1,y_2,..,y_n$ by K-means which k=2.*
5. *Let A_s,A_T be the submatrices of A. Recurse (Step 1-4) on A_s and A_T.*

ii. Merge phase

For a large class of natural objective functions proposed by the merge phase can be executed optimally when the expected number of clusters (i.e. K) is specified at first. Alternately, they use the most obvious turning point of K-TSS curve to automatically determine the value of K . Many inner measurements of the clusters effectiveness are based on the conception of cohesion and separation. Cluster cohesion (i.e. SSE) is the sum of the weight of all links within a cluster. Cluster separation (i.e. SSB) is the sum of the weights between nodes in the cluster and nodes outside the cluster. In some cases, there is a strong connection between the cohesion and the separation. Specifically, the sum of SSE and SSB is equal to total sum of squares TSS. TSS is defined as follows:

TSS=SSE+SSB. They observing that most obvious turning point of the K-TSS curve can help us determine the expected number of clusters.

This concludes that K-Divide-and-Merge clustering algorithm (BNAK-Divide-and-Merge) based on the Divide-and-Merge, improves the efficiency and performance of the clustering.

- **A Modified K-means Algorithm for Noise Reduction in Optical Motion Capture Data** [37]:

A modification to K-means algorithm has been used for removing noise in multicolor motion capture image sequences. The proposed algorithm takes into account the nature of the motion capture images in terms of the number of data pixels normally clustered together and the acceptable degree of compactness of a data cluster. The modified K-means algorithm is used to clean up the noise embedded in the color regions in each image by creating clusters of pixels based on their relative spatial positions in the image. Following the classical K-means algorithm, the Euclidean Distance measure is used to determine which cluster a pixel belongs to. Each pixel is put into a cluster, which yields the minimum Euclidean Distance between the pixel and the respective centroid. The centroid of each cluster is changed iteratively by calculating its new coordinate as the average of the sum of the coordinates of the pixels in the cluster until it converges to a stable coordinate with a stable set of member pixels in the cluster. In each iteration, the memberships of each cluster keep changing depending on the result of the Euclidean Distance calculation of each pixel against the new centroid coordinates.

Classical K-means algorithm is modified upon the form of constraints on cluster size and cluster compactness. The value for the cluster size constraint is set just above the number of data points usually found in a noise cluster for the type of data at hand. The value for the cluster compactness constraint is set just below the minimum compactness of valid data clusters.

- **Automatic Cluster Number Selection using a Split and Merge K-means Approach** [38]:

This research address the problem of cluster number selection by using a K-means approach that exploits local changes of internal validity indices to split or merge clusters.

There split and merge K-means issues criterion functions to select clusters to be split or merged and fitness assessments on cluster structure changes.

Assume a set of data samples X ={$x_1,\ldots,x_N$} is given, C={c1,…,ck} being the cluster centroid, the optimization criterion in the research is given as $L = \sum_{i=1}^{N} x_i^T c_{yi}$ where $y_i = arg \max_{1 \le k \le K} x_i^T c_k$ the hard assignment of samples to cluster is denoted as set y={$y_1,\ldots,y_N$}

- **Split and Merge K-Means**

- Require: X ,K, s(C), m(C), v(C)
- Ensure: C , Y

1: C = K-means (X_t , K)
2: Repeat
3: c_s = s(C), $X_s = \{x_n | y_n = s\}$
4: $\{c_i | c_j\}$ = K-means (X_s, K = 2)
5: if v(C) > v(C/c_s U $\{c_i | c_j\}$) then $C=C/c_s$ U $\{c_i | c_j\}$
6: until |C| is not changing
7: repeat
8: ci, cj = m(C)
9: $Y_j = Y_i$, $C = C/c_j$.
10: if v(C) > v(C/c_j) then
11: $C = C/c_j$
12: until |C| is not changing
13: C = K-means (X_t , C)

This split and merge K-means creates an initial partitioning through a first K-means step with a predefined number of clusters. Afterwards consecutive split and merge steps are invoked where the changes on the cluster result are assessed using some internal validity measure v(C) like the Bayesian Information Criterion (BIC). Those split and merge steps are repeated until changes no longer improve the fitness. At the end of the algorithm, an optional K-means step can further refine the results of the dynamic updates. Note that the input parameter K is optional and per default two, but the algorithm allows setting a preliminary expectation on the cluster number to reduce runtime. In order to reduce the number of splits and merges, algorithm also introduces a splitting criterion s(C) and a merging criterion m(C) for selecting the cluster to split or merge in a step. In this approach, s(C) selects the cluster with the lowest average data sample similarity. Similarly, m(C) selects the two most similar clusters as merging candidates. Researcher claims that split and merge K-means reaches the goal of providing a clustering structure

that dynamically selects its cluster number with an acceptable runtime and a favorable precision. In addition, this approach can be highly effective to generate an initial clustering result with an automatically detected number of clusters as well as in incremental applications where the given cluster hierarchy should be updated dynamically as new documents are added or old documents are removed. As a final remark, this split and merge approach seems to reach the goal of providing a clustering structure that dynamically selects its cluster number with an acceptable runtime and a favorable precision.

By the end of this subsection, the researcher observes that many of researches, that referred within this subsection or not refereed do not mention to inability k-means algorithms to cluster non-linearly separable datasets, which one of the main limitation of K-means algorithm. Accordingly, the researcher attempts developing a new algorithm to overcome a combination of K-means limitation such as: (i) noise or outlier which deteriorates the quality of clustering results (ii) initial centroids that have a direct impact on the formation of final clusters (iii) Inability to cluster non-separable datasets.

Chapter 3

3. Methodology and Design

Chapter 3

3. Methodology and Design

In this chapter, the researcher presents proposals related to K-means family of algorithms. Starting with identifying and explaining in depth the proposed seeding algorithm titled DIMK-means, which stands for "Distance-based Initialization Method for K-means clustering algorithm" which was developed to select a set of centroids that would result in a low cost clustering solution. Then the Chapter explores another proposed algorithm called DSMK-means which stands for "Density-based Split- and -Merge K-means clustering Algorithm" that have been developed to address stability problems of K-means clustering, and to improve the performance of clustering when dealing with datasets that contain clusters with different complex shapes and noise or outliers.

3.1 DIMK-Means "Distance-Based Initialization Method for K-Means Clustering Algorithm"

K-means algorithm is classified as a partition-based clustering technique, which is popular and widely used and applied to a variety of domains. K-means clustering results are extremely sensitive to the initial centroid; this is one of the major drawbacks of K-means algorithm. The researcher proposes a selection method for initial cluster centroid in K-means clustering instead of the random selection method. The research provides a detailed performance assessment of the proposed initialization method over many datasets with different dimensions, numbers of observations, groups and clustering complexities.

3.1.1 The Effect of Random Selection of the Initial Clusters' Centroids

Selection of initial centroids in K-means algorithm have significant impact on the results. The quality of K-means clustering results depends heavily on the manner of initialization. If this done incorrectly, things could go horribly, wrong. In this sub section,

we will illustrate by examples that choosing different starting point values lead to different clusters with different error values.

The first example shown in the Figure 3.1, which shows the results of running the K-means clustering algorithm on dataset with input parameter (k=2). This simple example shows that the position of starting point "initial cluster centroids" is important when trying to determine the best representation of clusters. When comparing Figures 8.1 and 8.2 visually; it can be determined which of the two clustering is "better", clusters in the second case "Figure 8.2" has better results as it could include all the points in each cluster while in the first Figure one of the points from the right cluster were included in left one. In addition, Figure 8.2 has lower values of objective function E than the first clustering result in "Figure 8.1".

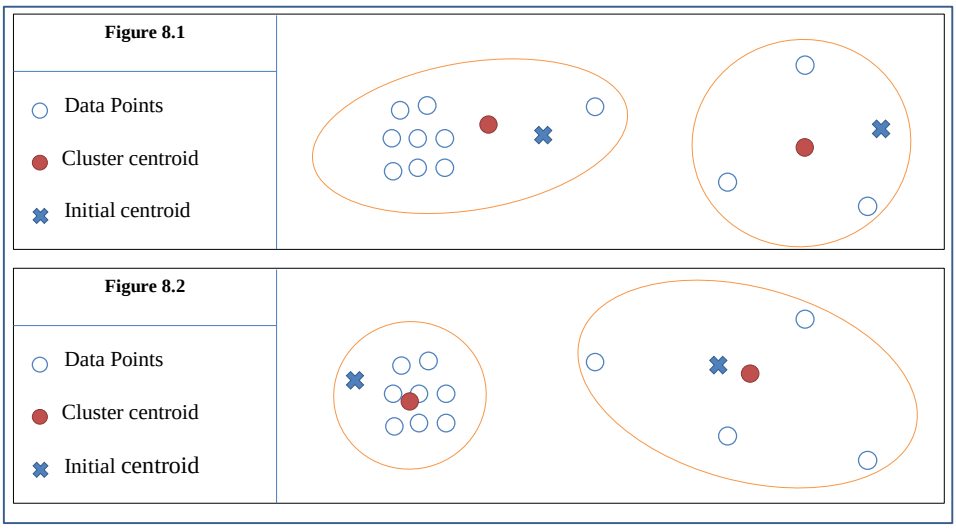

Figure 3.1: Example 1 show Initial centroid effects on K-means result.

In the second example shown in the Figure 3.2, a dataset is supposed to consist of N points in five tight clusters of some tiny radius arranged in a line, with some large distance D between them. These artificial datasets distribution is shown in Figure 3.2.1. If the input parameter k=5 to initialize K-means then five centroids are selected at random from the data. There is some chance that we would end up with no centroid from cluster 5, two centroids from cluster 3, and one centroid each from clusters 1, 2, and 4. These artificial datasets are shown in Figure 3.2.2. After the first iteration of K-means, all points in clusters 1 and 2 will be assigned to the leftmost center. The two centers in cluster 3 will end up sharing that cluster. In addition, the centers in clusters 4 and 5 will move roughly to the centers of those clusters. In this example, the results of K-means algorithm

shown in Figure 3.2.3 are bad as it merges the two most left clusters together and splits the middle one into two different clusters.

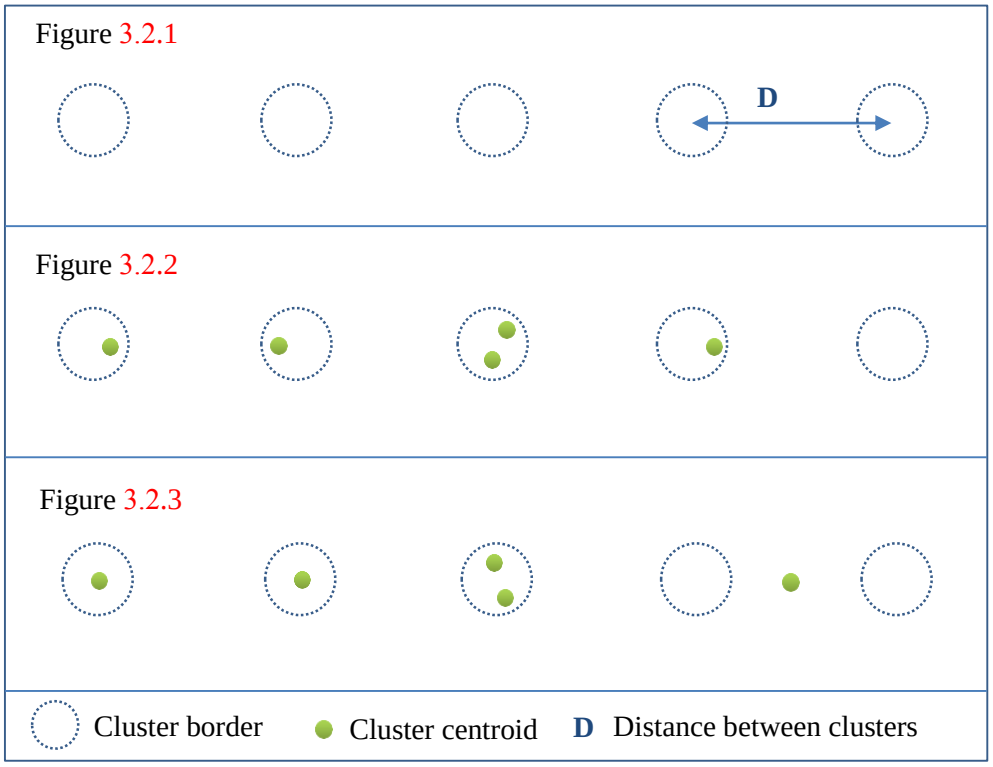

Figure 3.2: Example 2 show initial centroid effects on K-means result.

The above two examples show clearly how important the initialization process is and its effects on the results of K-means algorithm, which concludes that the selection of the initialization centroids is crucial. Most of developed algorithms to solve the initialization process sensitivity suffer from high computational complexity and therefore do not have strong scalability.

3.1.2 Proposed Method

First of all, It is well known that selection of the first centroids when they are far apart and each centroid belongs to different cluster has several benefits: [i] Decrease computation amounts, [ii] Optimize algorithm performance by minimizing the objective function of K-means algorithm which leads to better results.

This research proposes a new simple and scalable method for the initialization process in K-means; this method starts by choosing random initial centroid then some calculations are performed to decide whether the point is suitable to be considered as a first initial centroid or not. Such decision is based on the process of computing distances

between the selected centroid and other points within the dataset. The researcher uses two types of measuring distances between points "Euclidian or Manhattan" Because of the different nature of data. Figure 3.3 gives a general overview to the mechanism of the proposed method to calculate the best initial centroids for K-means algorithm.

To explain the example in Figure 3.3: Suppose we have a dataset as it occurs in figure and the value of K=3 which represents the number of clusters. DIMK-means will start by selecting first centroid randomly (*suppose the point shown in Figure 3.3 "iteration j" at the bottom right part as to be selected point*). After selecting the point, the algorithm is to compute distances among the random selected point and the rest points in the dataset. Then the values of $(\varepsilon, \varepsilon')$ calculated Based on the values of the distances calculated in the previous step, the algorithm decides that first selected point is not suitable and not good enough to be considered as basis for calculating the acceptance center because of the value of $\varepsilon'$is greater than ε. The algorithm still trying to select the first suitable centroid and the ε' is greater than ε , so a new point selected randomly as shown in Figure 3.3 "*iteration j+1*" and the current one ignored and is considered noise; because it is relatively far from the other points in the dataset. The algorithm repeat the previous steps on the new point and the values of $(\varepsilon, \varepsilon')$ calculated, because of the value of ε' is less than ε the algorithm decides to calculate the mean value of the nearest points to the selected point, which are shown in the Figure 3.3 "*iteration j+1*" inside the black-dotted line. The resulting mean value is considered as the first acceptance initial centroid ($\mathbf{C_1}$). Then those nearest points are ignored and a new farthest point is selected as shown in Figure 3.3 "*iteration j+2*" where the selected farthest point located at the left side. The algorithm repeats the steps on the farthest point and ε' is greater than ε and there is one acceptance centroids, then algorithm choose the closest point from the current point and repeat the steps of algorithm while current point will be considered as a noise. Because of the value of ε' is less than ε the algorithm decide to calculate the mean value of the nearest points to the selected point, which are shown in the Figure 3.3 "iteration j+2" inside the black-dotted line located at the left side. The resulting mean value is considered as the second acceptance initial centroid (C2). The same steps are executed again and the third acceptance initial centroid (C3) is calculated as shown in the Figure 3.3 "iteration j+3". All initial centroids are selected, and then the standard K-means is applied to the whole dataset.

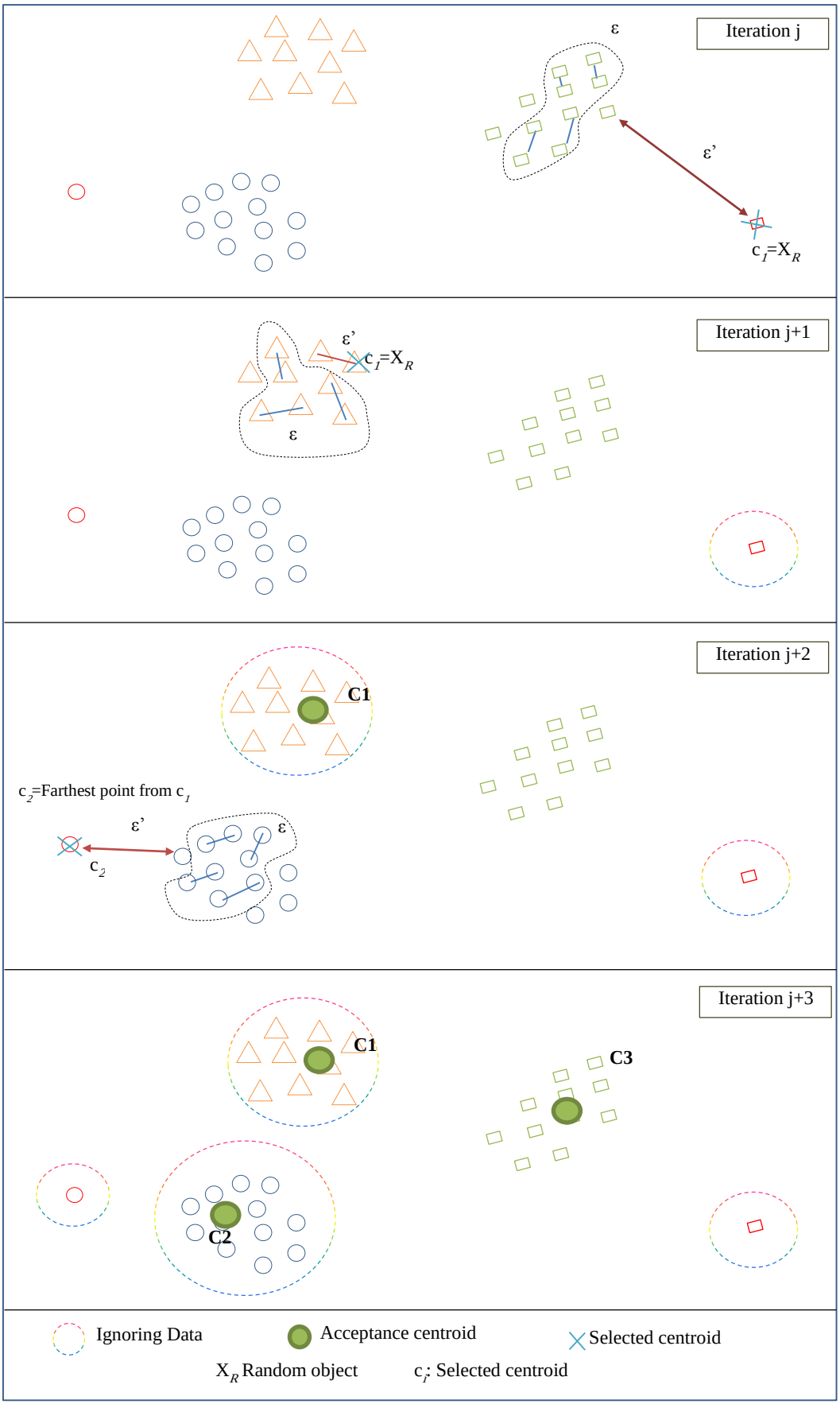

Figure 3.3: shows the operation of selecting candidates of the initial centroids from artificial dataset using DIMK-means.

Assumption I: *The number of objects in a cluster is close or equal to the number of objects in other clusters.*

This assumption is based on the fact that K-means algorithm always get better results with datasets which are similar in density and close in the objects number in each cluster. Therefore, this assumption is valid for a large number of datasets.

Computing the distances between the selected point and the remaining points is the backbone of this method because the distance values between the selected centroid and its nearest point is used to calculate the value of ε' and is compared with the value of ε which is equal to the mean value of the distances between each pair of N points.

Determine the number of the closest points to the selected centroid depending on the Assumption I, where the number is equal to 80% to 90% of the number computed from dividing the total number of dataset objects divided by the number of clusters given from the user. If the first selected point was noise; i.e. ε'> ε; this point is ignored and another point should be selected randomly as initial centroid until the first centroid is found. Then; the next centroid should be selected as the farthest points from the first centroid. If the second selected point was noise; it is ignored and its closest point is selected as centroid which in turn should be tested if it is noise or not. Then, the mean value of N number of the closest point to the current centroid is saved as the first accepted centroid which is ignored in the following computations. This method is repeated until the required number of centroids is identified.

3.1.3 DIMK-means Steps:

The proposed clustering algorithm "DIMK-means" consists of five main steps, which are as follows:

Suppose that we are going to partition $X=\{x_1,x_2,...,x_N\}$ which is a dataset with N number of objects, and K is an input parameter equal to number of clusters.

1. **Select First Centroid Randomly:**

   DIMK-means starts with selecting a random point to be used in computing the first centroid. Which in turn is used to calculate other centroids by computing the distances among the dataset points.

2. **Calculate distances from the selected point to the other points:**

   The base of this algorithm is to compute distances among the random selected point and the rest points in the dataset. Researcher choose distance calculation algorithm

"Euclidean distances" to be applied, that distance is special because it conform to our physical concept of distance.

3. **Calculate the values of (n, ε , ε′):**

   Based on the values of the distances calculated in the previous step, the algorithm decides if the selected point is suitable and good enough to be considered as basis for calculating the acceptance center or not.

   The variables (n, ε , ε′) mainly depend on the values of the distances between the selected point and the other points in the dataset, these variables are essential to make a decision whether this point is suitable or not.

   The used variables are defined as follows:

   - ***n:*** *Is the minimum expected number of points located in each cluster that belongs to a particular dataset and closest to the selected point, Depending on "assumption I", the value of n is usually equals a certain percentage of the total number of points divided by K of centers.*
   - ***ε :*** *Is the average value of distances between each pair of n points, these n points are the nearest points to a selected point.*
   - ***ε′***: *Is the distance between the selected point in the first step and the nearest point.*

4. **After calculating (n, ε , ε′), the selected point is checked whether it is valid to be used to calculate the acceptance centroid or not.**

   Determining if the selected point is appropriate or not is based on the values of variables (n,ε ,ε′) as follows:

   a. *If ε′ is greater than ε and the first initial centroid is not selected yet, then another point should be selected randomly and the current one should be ignored and is considered noise because it is relatively far from the other points in the dataset.*
   b. *If ε′ is greater than ε and there is one or more selected centroids then choose the closest point from the current centroid. The current point will be considered as a noise, while the new closest point will be used to calculate a new centroid as in steps 1 through 3 by calculating the distances and finding new values for n, ε and ε′.*

c. *If ε' is less than ε then calculate the mean value of the nearest n points to the selected point. The resulting mean value is considered as the acceptance initial centroid. Then those n points are ignored and a new farthest point is selected and step 2 is repeated until all centroids are selected.*

5. **After all centroids are selected run K-means with selected initial centroids parameters.**

   After selecting all initial centroids, the original K-means is applied to the whole dataset.

3.1.4 Advantages and Limitations of DIMK-Means Algorithm

- **Advantages:**
  a. The algorithm is not difficult to implement.
  b. The algorithm does not require any additional parameters more than the standard K-means algorithm.
  c. The algorithm makes K-means less sensitive to noise.
  d. The performance of K-means algorithm with the proposed initialization method "DIMK-means" is more effective and converges to more accurate clustering results than those of the random initialization method.
  e. The proposed method has substantially outperformed the standard K-means in terms of speed; It is true that the proposed initialization method needs more time than random initialization method but the initial centroids selected by the proposed initialization method are very close to the true clusters' centroids, thus reducing the rest standard K-means computations.

- **Limitations:**
  1. DIMK-means algorithm did not reduce the number of parameters needed.
  2. When a number of objects in a cluster is not close to the number of objects in the other clusters, DIMK-means gives a similar or slightly better performance than the standard K-means. This is not a big problem considering that K-means algorithm always gets better results with datasets, which are similar in density and size.

3.1.5 DIMK-means Algorithm Pseudo-Code

Suppose that we are going to partition $X=\{x_1,x_2,...,x_N\}$ which is a dataset with N number of objects, and K is an input parameter equal to number of clusters.

Algorithm 3.1 : DIMK-means

Purpose: Clustering dataset

Input: $X=\{x_1,x_2,\ldots,x_j\}$ (set of entities to be clustered)
K (number of cluster)
MaxIters (Limit of iterations)

Output: $C=\{C_1,C_2,\ldots,C_K\}$ (set of cluster centroids)
L= (set of cluster labels of X)

Procedure

1. Choose an initial centroid $ci = x_r$, where $0< i \leq K$ and x_r random from X.
2. Compute the distance between selected centroid ci and each point in X, and then sort the data points based on the resulted distances.

$$D = d(c_i, x_j)$$

Where D: typically is chosen as the Euclidean distance, $0 < j \leq N$.

3. Get a subset of the sorted data with a number of points equal to N

$$n = \text{CeilEven}\left(\frac{N/k}{\sigma}\right)$$

Where n: number of data most close to the selected centroid c_i, σ is a double number $1< \sigma \leq 2$, and CeilEven is a function that rounds a double number up to the nearest even integer.

4. Compute the average distance between each pair of n points

$$\varepsilon = \left(\sum_{\substack{m=0, \\ m=m+2}}^{n} d(x'_{m+1}, x'_{m+2}) \right) \Big/ \frac{n}{2}$$

$$\varepsilon' = d\,(c_i\,, x^*)$$

Where x' represent the closest data points to ci, while m is incremented by 2, and x^* is the closest point to ci.

5. If { $\varepsilon' > \varepsilon$ and i=1}; ignore ci and go to step 1 to select a new ci
6. If { $\varepsilon' > \varepsilon$ and i > 1}; ignore ci, select a new ci with value equal to the closest point to the previous ci; and go to step 2.
7. Choose the next centroid c_i+1 to be the farthest point from ci.

8. The mean value of n points closest to ci is identified as the centroid and is saved as "acceptance centroid C_i ".

$$C_i = \left(\sum_{m=0}^{n} x'_m \right) / n$$

Where C_i: represent the mean value of the closest points to ci.

9. Ignore n points, which are the closest to ci.
10. Go to step two with value of $c_i = c_i+1$.
11. Repeat steps until a total of K centroids are chosen.
12. Run K-means Algorithm with selected centroids.
13. End;

The following Figure 3.4 show flowchart that exhibits the process of DIMK-means and explains the abovementioned pseudo code:

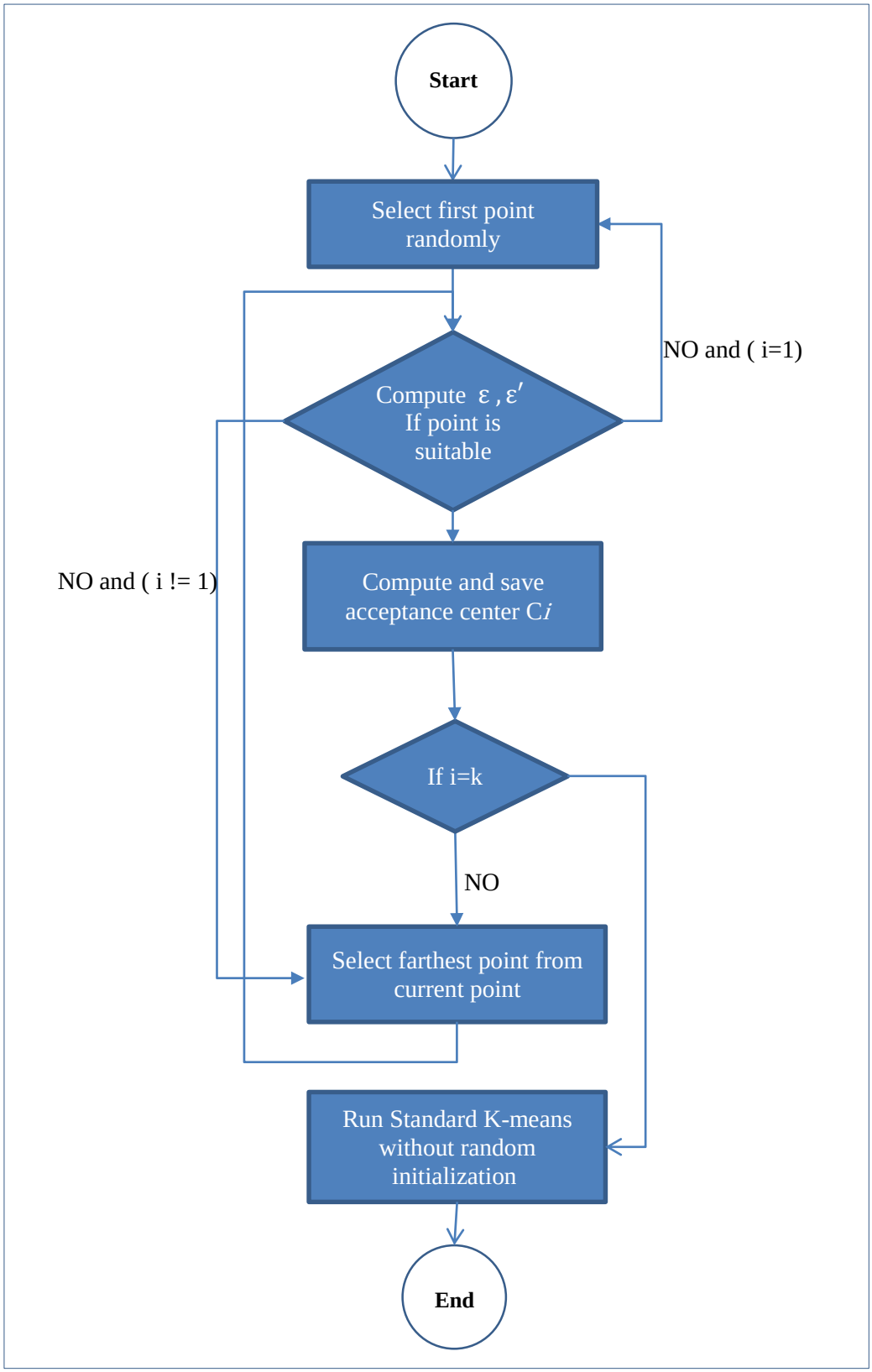

Figure 3.4: Flowchart of DIMK-means algorithm.

3.2 DSMK-Means "Density-Based Split-And-Merge K-Means Clustering Algorithm"

The K-means algorithm is a simple and fast clustering technique that exhibits the problem of merging some clusters which are close together. In addition to that, the algorithm generally suffers from unsatisfactory accuracy when the dataset contains clusters with different complex shapes, sizes, noise and outliers. In this work, researcher addresses these problems by combining split and merge strategy and density clustering techniques. The proposed density-based split and merge K-means algorithm comprise of two parts, the first one depends on density to decide if the cluster to be split or not, and distance to decide if the clusters to be merged or not.

If the first part was not applicable, then the algorithms applies the second part which tackles noisy data and depends on density to identify noisy objects or points in a dataset. The next section explains this procedure in more details.

Using density with split and merge techniques in this algorithm makes the proposed algorithm capable of detecting clusters with different complex shapes. Furthermore, density technique helps in discovering noise or outlier. This gives the proposed algorithm higher accurate results than the standard K-means algorithm when applied on datasets containing large numbers of objects, clusters with different shapes and/or clusters containing noise objects.

3.2.1 Performance of Standard K-Means

This subsection discusses a set of experiments on K-means algorithm with different datasets. These experiments illustrate the ability of K-means algorithm to find the true cluster, as clarifying the strengths and weaknesses of algorithm is the end purpose of these experiments.

To establish practical applicability of K-means algorithm, its performance was tested on a number of artificial and real world datasets. Those datasets contain clusters with different complex shapes, densities, sizes, noise and outliers. The main purpose is to show how K-means work with this type of datasets. It was experimented on two different types of datasets which are: Artificial (Ground Separation, document Sim, and Rnoisy) and real datasets (Web Log, Image Extraction). These datasets are described in depth in Chapter 5.

The next paragraphs illustrate researcher observations on the results of standard K-means algorithm on all previous datasets.

❖ **Interpreting Results of K-means with Ground_Separation dataset:**

In many clustering analysis problems, one would like to extract structure from cluttered background. This is the case in the Ground_Separation dataset. In such cases, it is easy to predict that K-means will not get accurate results, due to their requirement to partitioning all the input data. To illustrate this point, consider the Ground_Separation dataset shown in Figure 3.5, which contains a dense central cluster of random points surrounded by equally distributed clutter points (the "background") and there are four extra clusters around the ring cluster. As expected, on these data, K-means failed as it split the central group into multi pieces.

This experiment was running many times on Ground_Separation dataset as shown in Figure 3.5 The main feature of this dataset is that it contains different structurally clusters, one is compact, and the other with extended structure. Here, K-means produces in-accurate results, as shown in Figure 3.6.(A, B, C, and D). After running the algorithm lots of times with these datasets, the results were bad every time. Researcher noticed that some parts of the ring-shaped cluster were classified with disparities between five or six different clusters, even though all the points forming the ring belong to one cluster. These results shown in Figure 3.6.(A,B,C and D).The most important general observation is that centroids of clusters obtained from K-means results, which plotted as x on Figure 3.6, is always not in a dense area.

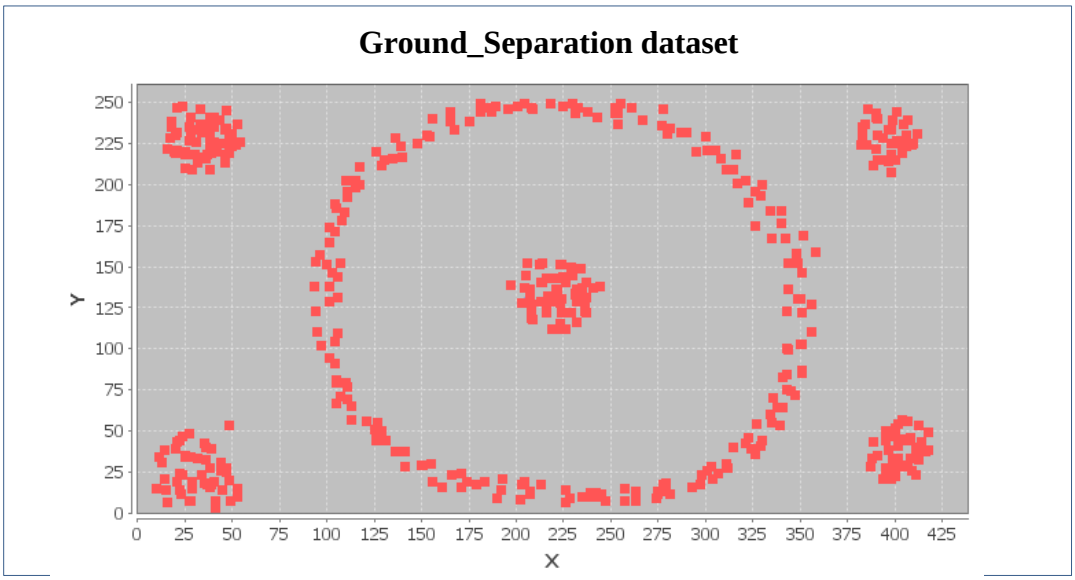

Figure 3.5: plot points belong to Ground_Separation dataset.

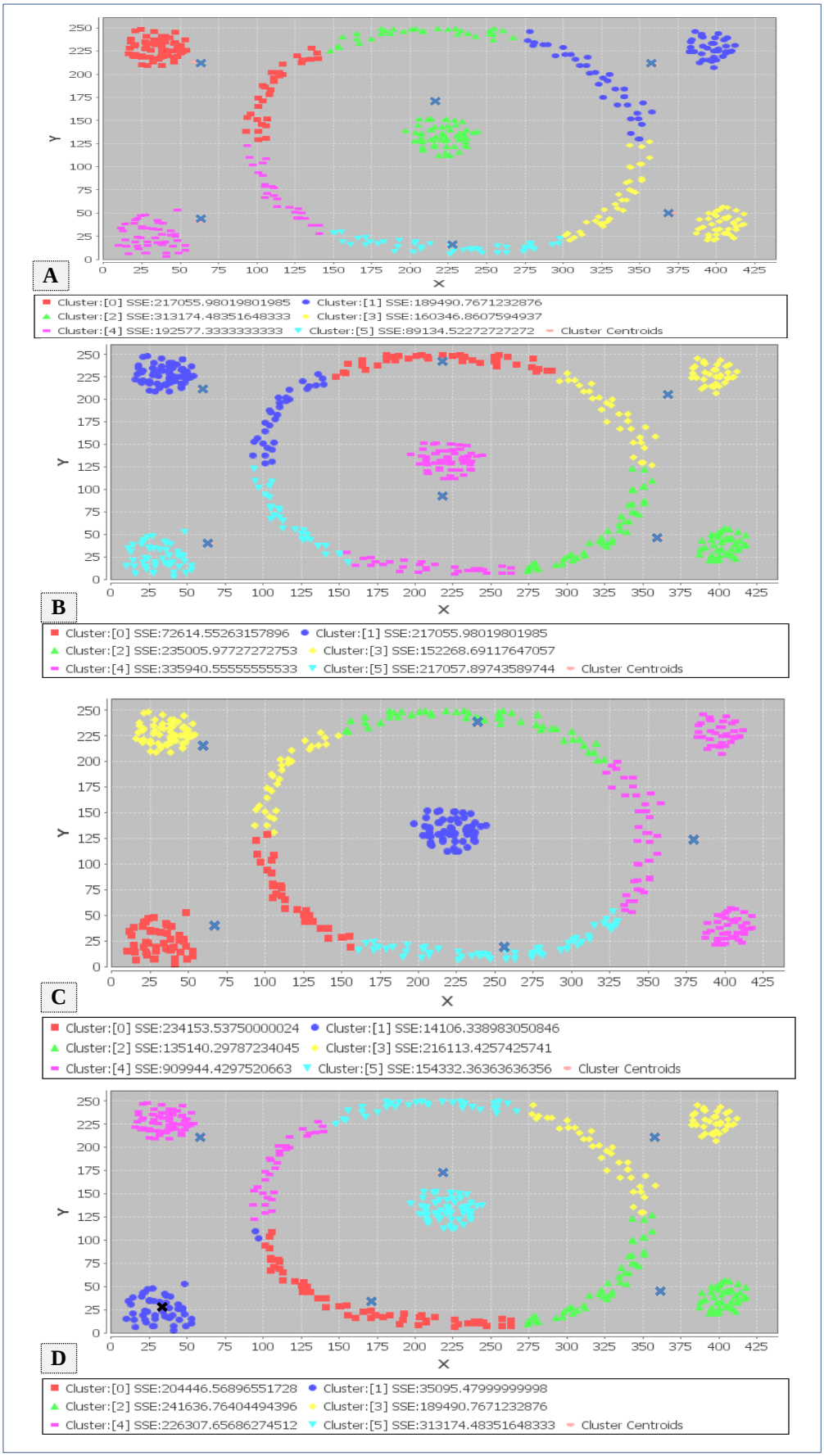

Figure 3.6: Low accurate results obtained with standard K-means algorithm with (Ground_Separation dataset).

❖ **Interpreting Results of K-means with Weblogs dataset:**

Using K-means to get real work done means running the algorithm lots of times on different types of datasets. This time we test the algorithm performance on real datasets "Weblogs". In such cases, it is easy to predict that K-means will not work well, due to their different cluster shapes in weblog dataset. Weblogs datasets that shown in Figure 3.7 is comprised of three clusters, which include outliers, the two clusters on both ends, have a sphere shape while the third cluster, in the middle, contains a large number of objects, which extends horizontally as "Gaussian structure". Figure 3.8 shows the results of standard K-means applied on weblogs dataset, where it is easy to observe with naked eye the low accuracy of results. In Figure 3.8 (A, and B) the curve with a dotted line in black represents bad results as one true cluster was merged with part of the cluster in the middle. Furthermore, in Figure 3.8 (C); the results were very bad because parts of the middle cluster "cluster with Gaussian structure" was merged with two other existing clusters which marked by curves with a dotted line in black and red.

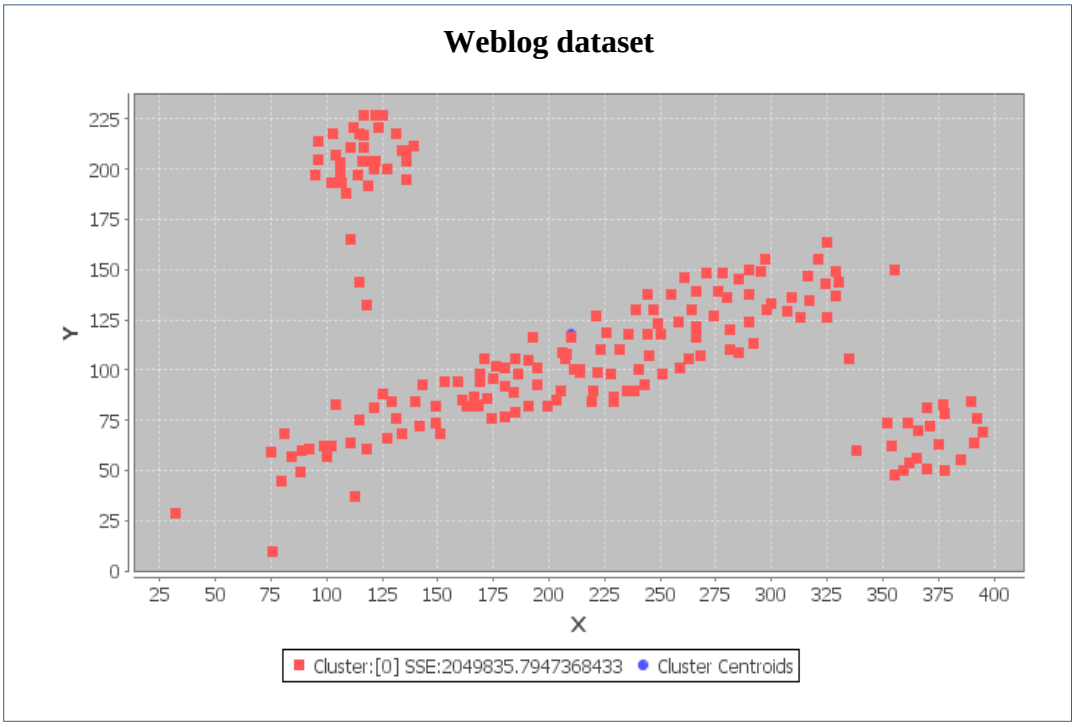

Figure 3.7: plot points belong to Ground_Separation dataset.

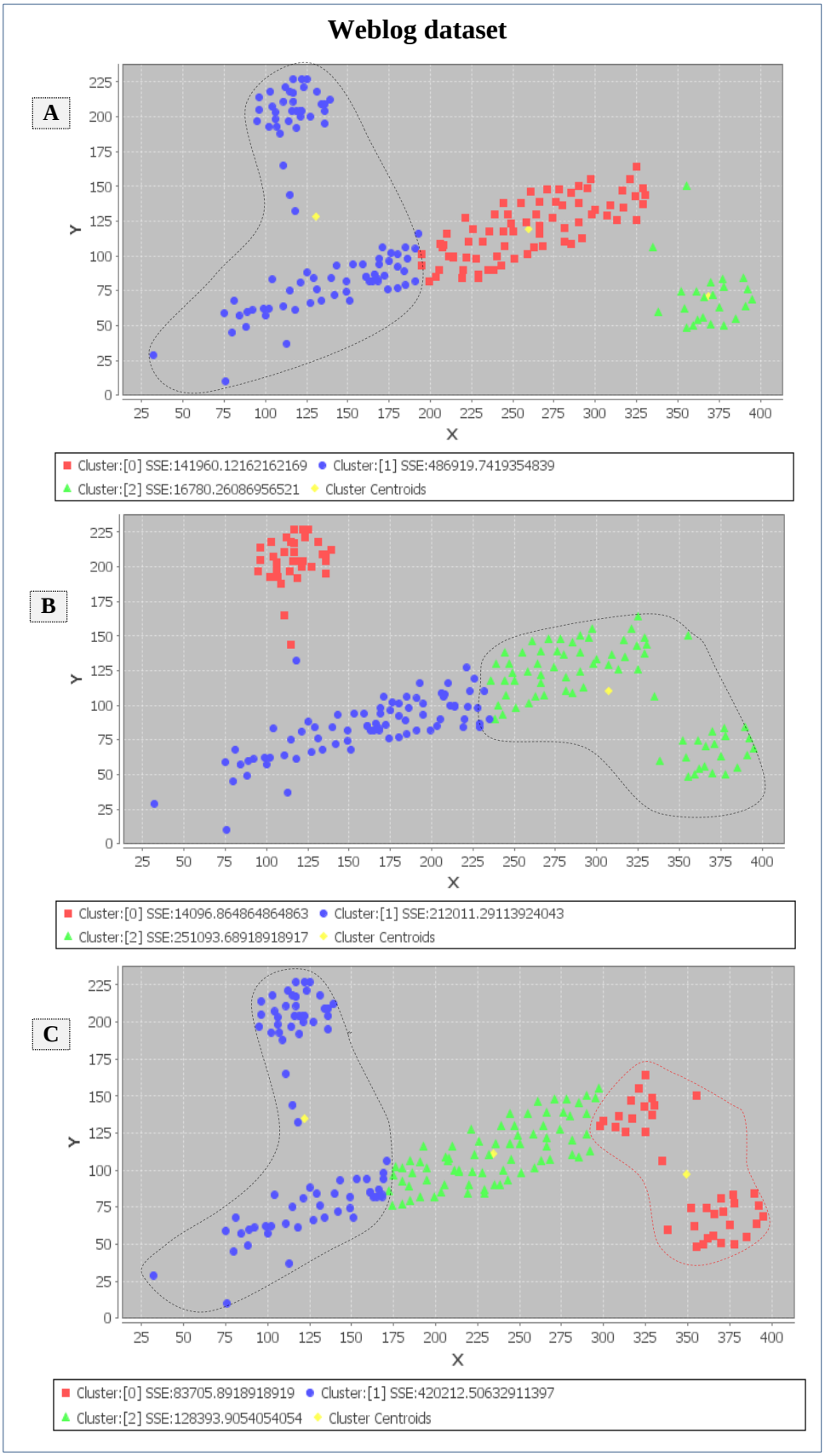

Figure 3.8: Low accurate results obtained with standard K-means algorithm with (Weblog dataset).

❖ **Interpreting Results of K-means with Image_Extraction dataset:**

The researcher applied K-means on Image-Extraction dataset, and as the previous experiments, it was easily noticed that the results were inaccurate. The researcher observed during the tests that the result always took approximately the same shape. Figure 3.9.A represents the plot of the datasets points on two-dimensional space. While Figure 3.9.B represents K-means result with the same dataset. Like the previous experiments, the observation was that bad clusters' centroids obtained from K-means results were not in dense area.

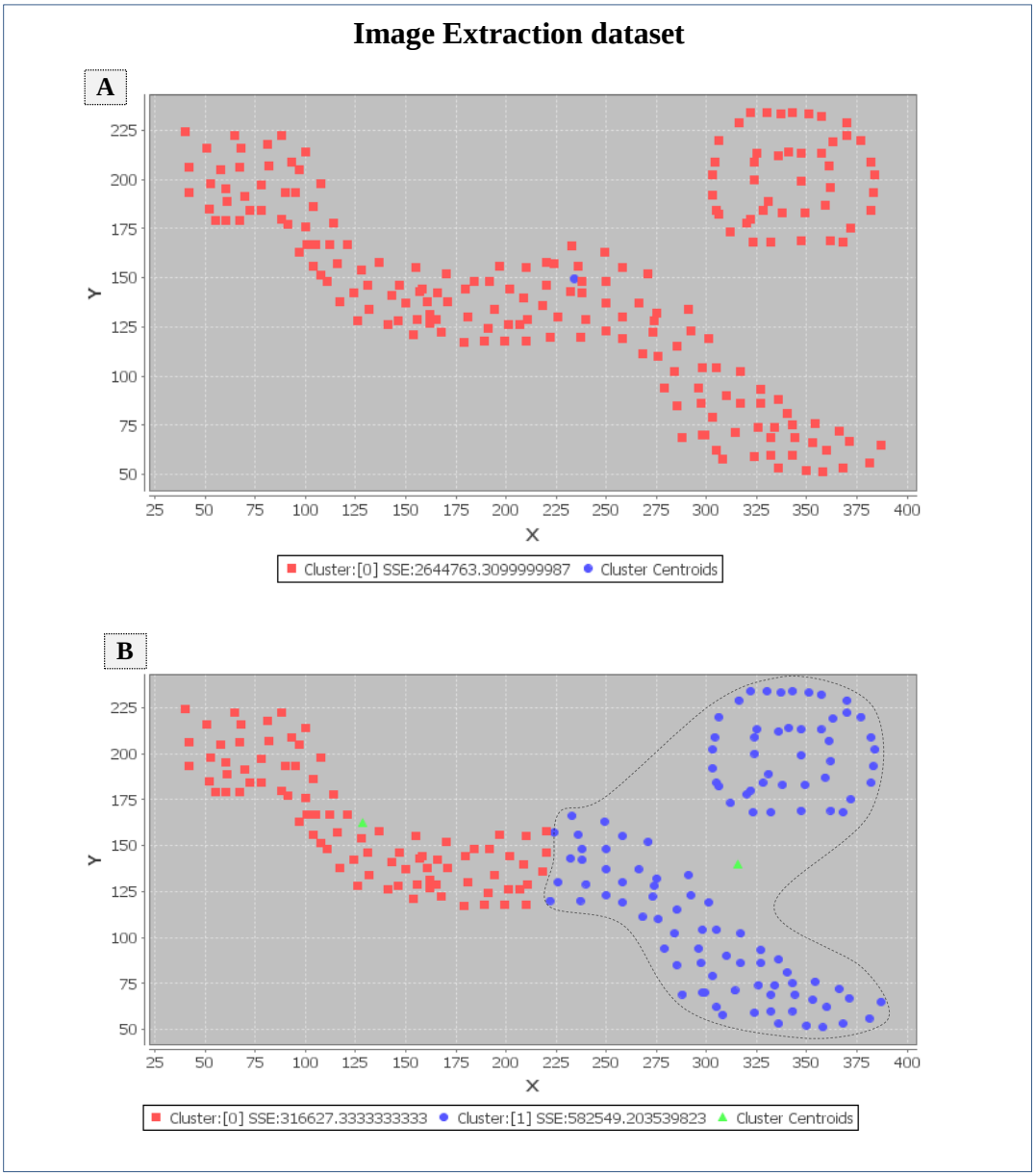

Figure 3.9: Low accurate results obtained with standard K-means algorithm with (Image Extraction dataset).

❖ **Interpreting results of K-means with Rnoisy dataset:**

K-means was applied on Rnoisy dataset and the results were sometimes of high accuracy and other times with bad accuracy, which convinced the researcher that K-means has unstable results when applied on datasets similar to Rnoisy datases, which contain many noisy points.

K-means algorithm high accurate results obtained when applied on Rnoisy dataset are shown in Figure 3.10. It is clear in Figure 3.10 that the data contain many noisy points which K-means algorithm is very sensitive to. The researcher observed during the tests that the shape of clusters in results takes different forms in each time. Figure 3.11(A) summarizes the results and it is easy to observer with naked eye how low accurate the obtained results are. Curves in Figure 3.11(A) show the bad results area were red-dotted line shows that one true cluster has been split into two clusters "blue and red", while the black-dotted line shows that two true clusters merged into one cluster "yellow". Many other bad results occur repeatedly In Figure 3.11(B.C.D). Finally, researcher observed that noisy points are always not in dense area.

This observation was the basis the researcher depended on to develop new ways to overcome the weakness of K-means algorithm when working with noisy datasets.

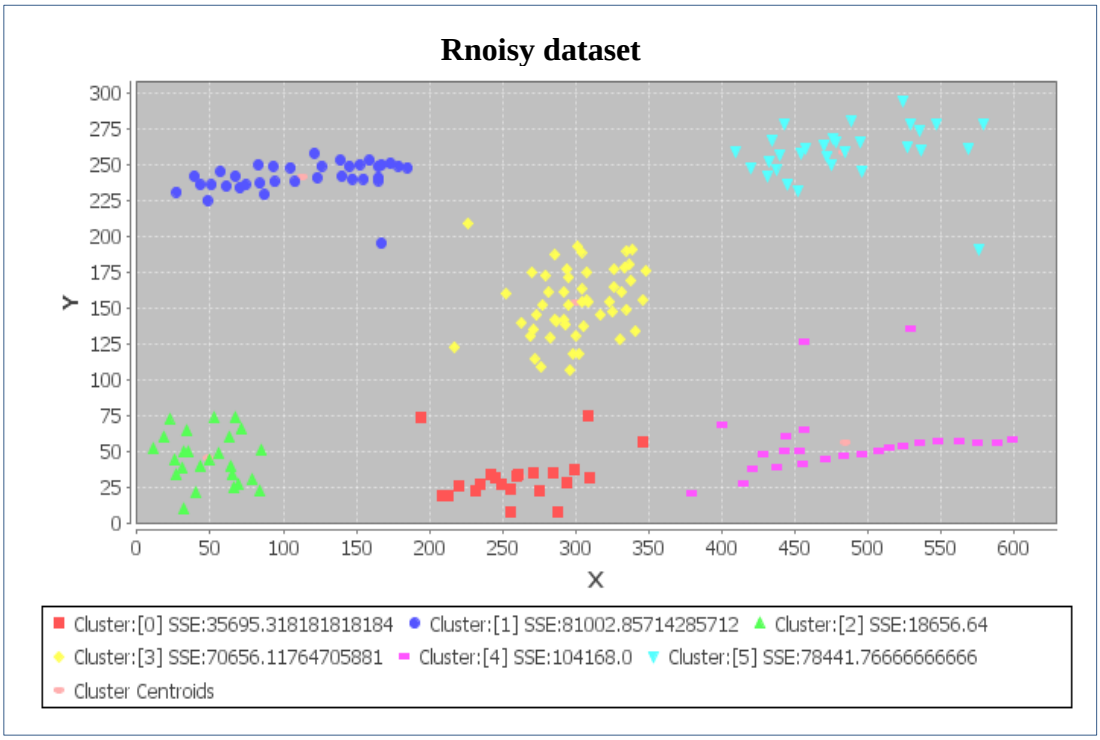

Figure 3.10: High accurate result obtained with standard K-means algorithm with (Rnoisy).

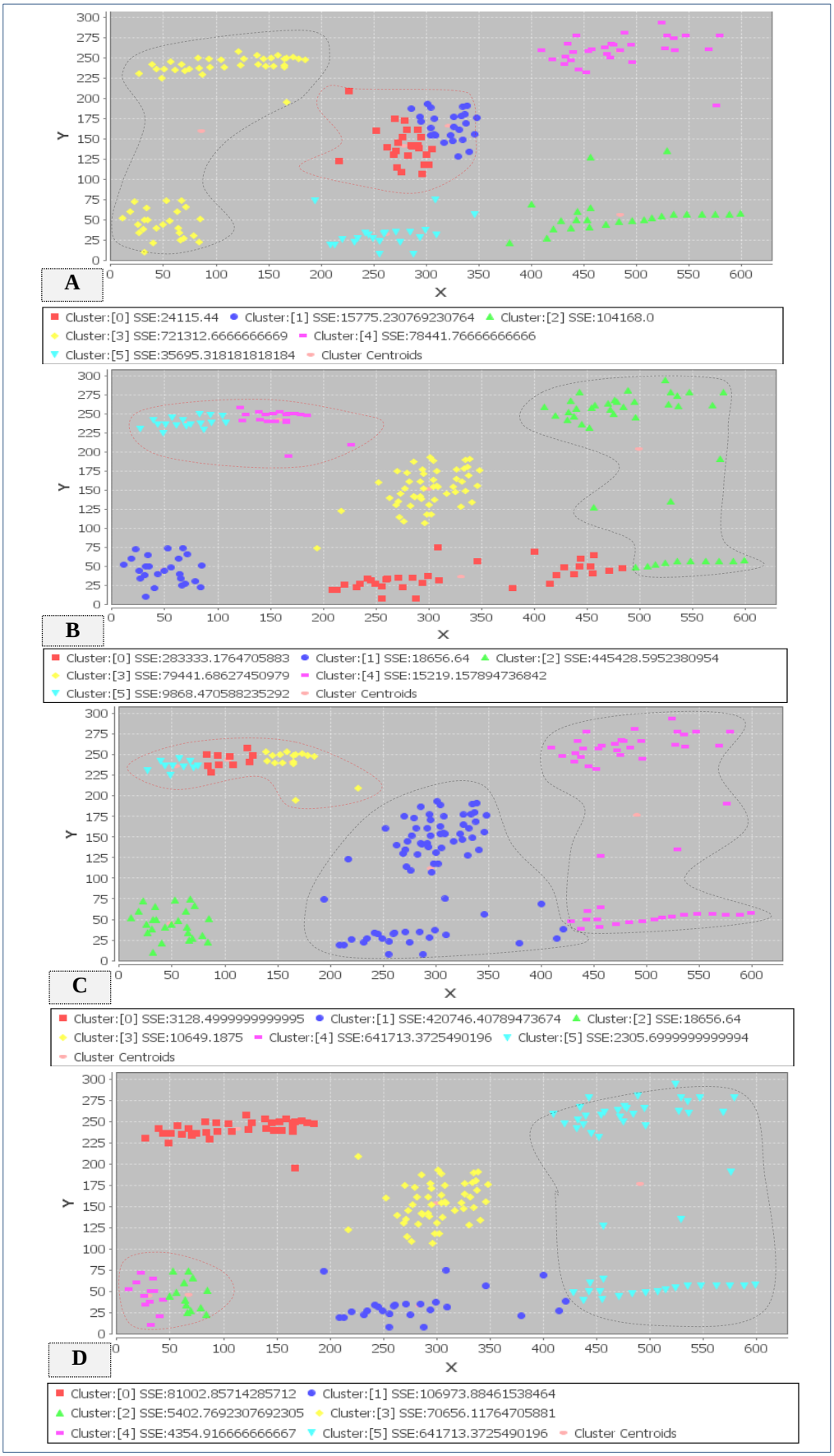

Figure 3.11: Low accurate results obtained with standard K-means algorithm with (Rnoisy dataset).

❖ **Interpreting Results of K-means with Document_Sim dataset:**

By running K-means, multiple times with Document_Sim dataset that contains number of noise points larger than Rnoisy dataset, in many times the results were unacceptable with low accurate results. In rare times, standard K-means algorithm obtained high accurate results when applied with Document_Sim dataset as shown in Figure 3.12.

To illustrate the unacceptable results "bad result" let us see Figure 3.13 (A.B.C.D). The Figure shows the bad results areas which are inside the red-dotted line. That line represents either areas where one true cluster has been split into multiple clusters or represents a cluster which is basically formed of noise points, while the black-dotted line shows that two or more true clusters are merged into one cluster. The noise points always are not in dense areas. This encourage the researcher to develop new ways to overcome these weaknesses of K-means algorithm.

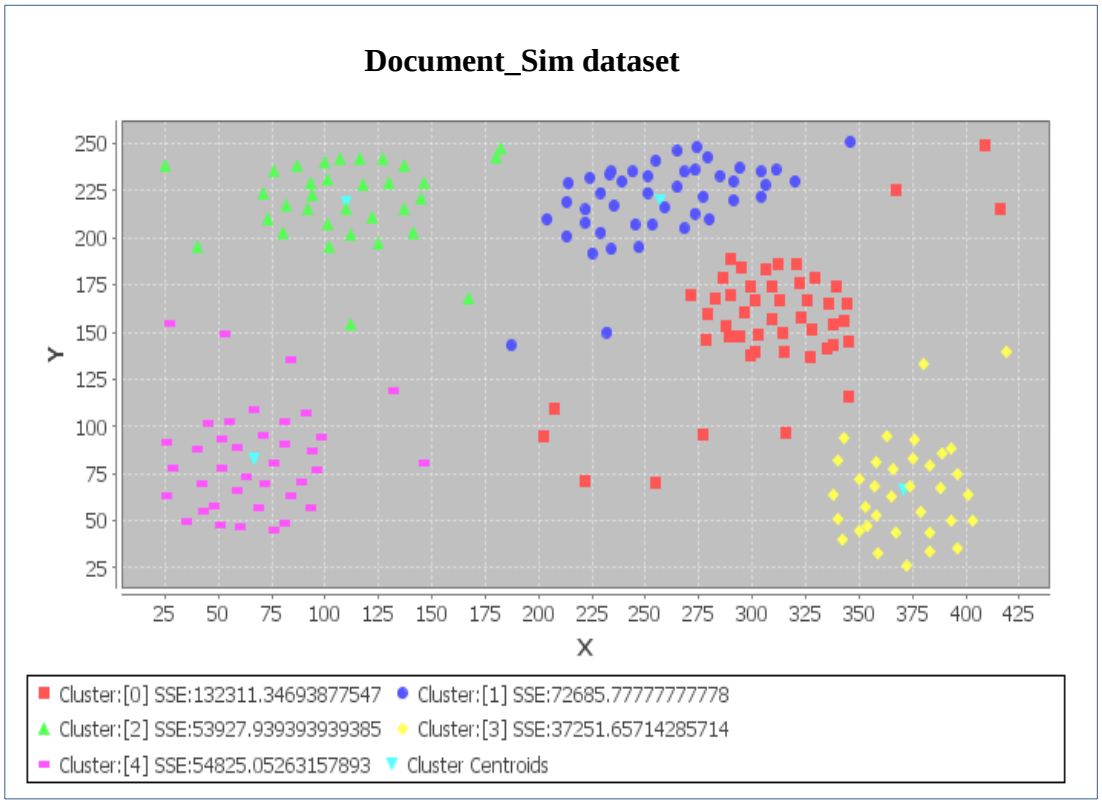

Figure 3.12: High accurate result obtained with K-means algorithm with (Document_Sim).

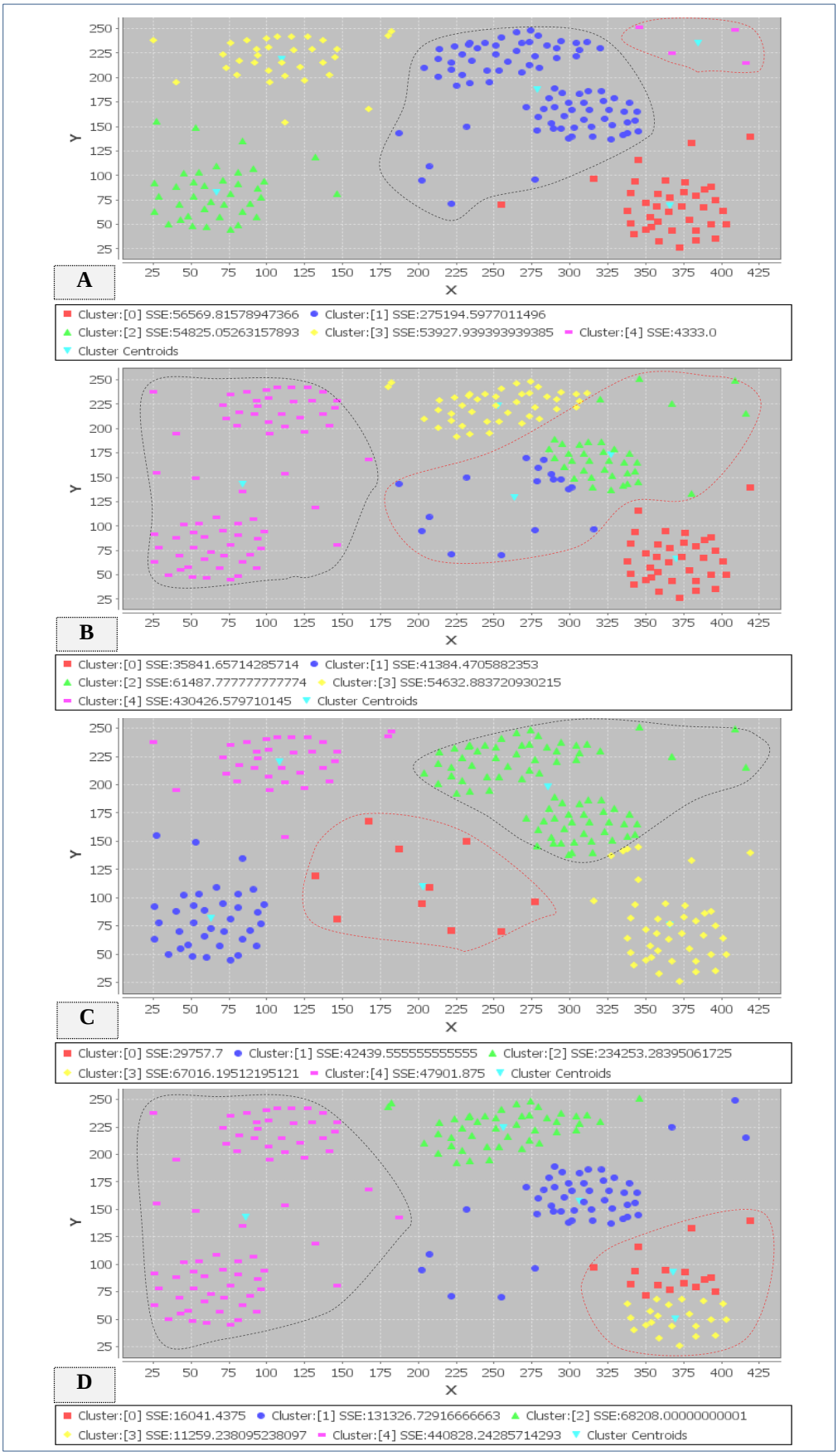

Figure 3.13: Low accurate results obtained with standard K-means algorithm with (Document_Sim dataset).

3.2.2 Proposed Solution

A discussion of the previous subsection experiments results shows the performance of K-means algorithm with different datasets with different behavior. Now researcher reviews the proposed ideas designed to overcome and solve major limitation and weaknesses of K-means algorithm. Generally, the algorithm suffers from unsatisfactory accuracy when the dataset contains clusters with different complex shapes, sizes, noise and/or outliers.

Based on the observation from the previous experiments where K-means merged true clusters, the resulting cluster centroid was -most of the time- not located in a density unit as it is locate between multiple true clusters. This observation was a result of the fact that K-means algorithm gets low accurate results when working with datasets contains clusters with different complex shapes. So, the researcher proposes to apply Split and Merge technique to overcome such limitation.

Another observation is the low accuracy of K-means algorithm when working with noisy datasets where noise or outliers always spread between datasets objects not in density unit. A proposed solution to overcome such limitation is by temporarily ignoring noisy objects which are not located in dense units, then rerunning standard K-means which is expected to give better results without the neglected noise. After that, re-include the neglected noisy objects to the nearest clusters.

The proposed algorithm includes solutions for cluster with complex shapes and datasets with noisy objects. The solution for the first problem is split and merge while the solution for the other problem is called anti-noise. This algorithm is applied on the results of standard K-means starting with checking if all the clusters' centroids are located in density units, anti-noise solution is applied, but if one or more centroids are located in non-density unit, then split and merge solution is applied.

The following cases explain in details how each solution is implemented:

- **Split and Merge Method:**

When applying standard K-means on datasets containing clusters with different complex shapes, some of the resulting clusters are either merged into larger clusters or split to smaller ones. First, in order to determine which method to apply we need to identify if the clusters have centroids in non-density units. So, Sum Square Error "SSE"

is computed for each cluster in standard K-means results where the cluster with smallest SSE value is selected, then we compute Epsilon "ε" which is the radius that delimitates the neighborhood area of a point by calculating the distance between the centroid of the selected cluster and the nearest point multiplied by two, which is the shortest suitable Epsilon "ε" distance that almost contain appropriate number of points. Then calculating "MinPts" which represent the minimum number of points that must exist in the ε distance. MinPts is equal to 0.75 of that number of points within ε radius (approximated to an integer number). The researcher takes 0.75 of the points to be considered as MinPts in order to exceed the varying density of clusters centroids.

In addition of previous reasons and based on experiments, the researcher found that multiplying the distance between the centroid and its nearest point by 2 is the most convenient and yields the best results most of the time, as well as determining MinPts by multiplying the number of points falling within ε radius by 0.75.

Second, each cluster centroids in the standard K-means results is tested to make sure it has a number of point equal to or greater than MinPts. If there is at least one centroid that has a number of neighbor points within ε" radius that is less than MinPts; then it is not in a density area, and we start the Split and Merge method, otherwise we use Anti-noise method as in case 2.

- ***Density-based cluster split:***

The splitting process is applied on clusters with centroids located in non-density units, each of those clusters is split into two new clusters. The resulting cluster centroids are tested to assure them all located in density units. The process of splitting is repeated until all the resulting cluster centroids are located in density units using the same ε and MinPnts calculated at the first run.

Splitting clusters into only two new sub-clusters instead of three or more is based on the fact that the possibility of having new cluster centroids in density unit in the least number of possible sub-clusters is higher than having such results in more than two sub-clusters.

A counter in increased by one each time a cluster is split, in order to keep record of how many split process were done to be used in the merge process.

- ***Single linkage based cluster merge:***

When the split process is finished, all clusters' centroids are in density units and the number of clusters is more that the number of clusters obtained from the standard K-means applied in the first step.

The merge process starts by creating "distance matrix" between each pair of clusters' centroids including all clusters within the dataset. The resulting distance matrix is *y*y* matrix where *y* is the number of all the clusters in the dataset including the clusters resulted after the split process. This matrix is used to identify the two most close cluster centroids with the dataset in order to check if they both belong to one true cluster. Single linkage "nearest neighbor or shortest distance" concept is applied for this purpose, where it calculates and finds the shortest distance between a pair of objects each of them is located in one of the selected closest clusters. Then, the merging will take place if at least one of the following conditions is true:

1: The distance between the two nearest points that belong to the clusters with closest centroids is less than or equals to ε.

2: The point in the middle between the selected pair of objects is checked if it is in a density unit and has a number of points equal or larger than MinPts within ε radius that belong to the closest clusters.

If one of the above conditions is fulfilled, then the two closest clusters are merged, then the distance matrix is calculated and the process is repeated as many times as the split process. Otherwise, the second shortest distance from the distance matrix is selected and the process is repeated.

At the end of this process, the number of resulting clusters is the same as the number of clusters resulted from the standard K-means where K is user parameter.

- **Expressing Split and Merge Method by Example *:***

One of the famous and complicated clustering analysis problems is to extract structure from cluttered background. This is the case, for example, with figure/ground separation and perceptual grouping like Figure 3.6 and Figure 3.14. The last figure, specifically Figure 3.14(A), shows results of standard K-means algorithm that gives very bad results as it split cluster in the center of Figure into two parts each part merged into

different clusters, one of these clusters plotted in red color with square shape, and the other plotted in blue color with circle shape.

When applying split and merge method on the same problem Figure 3.15, each of the two clusters obtained from standard K-means were split into three clusters on two runs, resulting in six new clusters instead of the original two. The split counter recorded four splits. In merge process, the two clusters in the middle were merged into one cluster, while the clusters in the ring were merged into one cluster through three merge processes. The process of DSMK-means clustering algorithm is explained in Figure 3.15. The final shape of the results is in Figure 3.14(B).

The results obtained from the split and merge method cannot be obtained from the standard K-means. As the applied method used the density and single linkage concepts combined with the standard K-means algorithm, which resulted in better results.

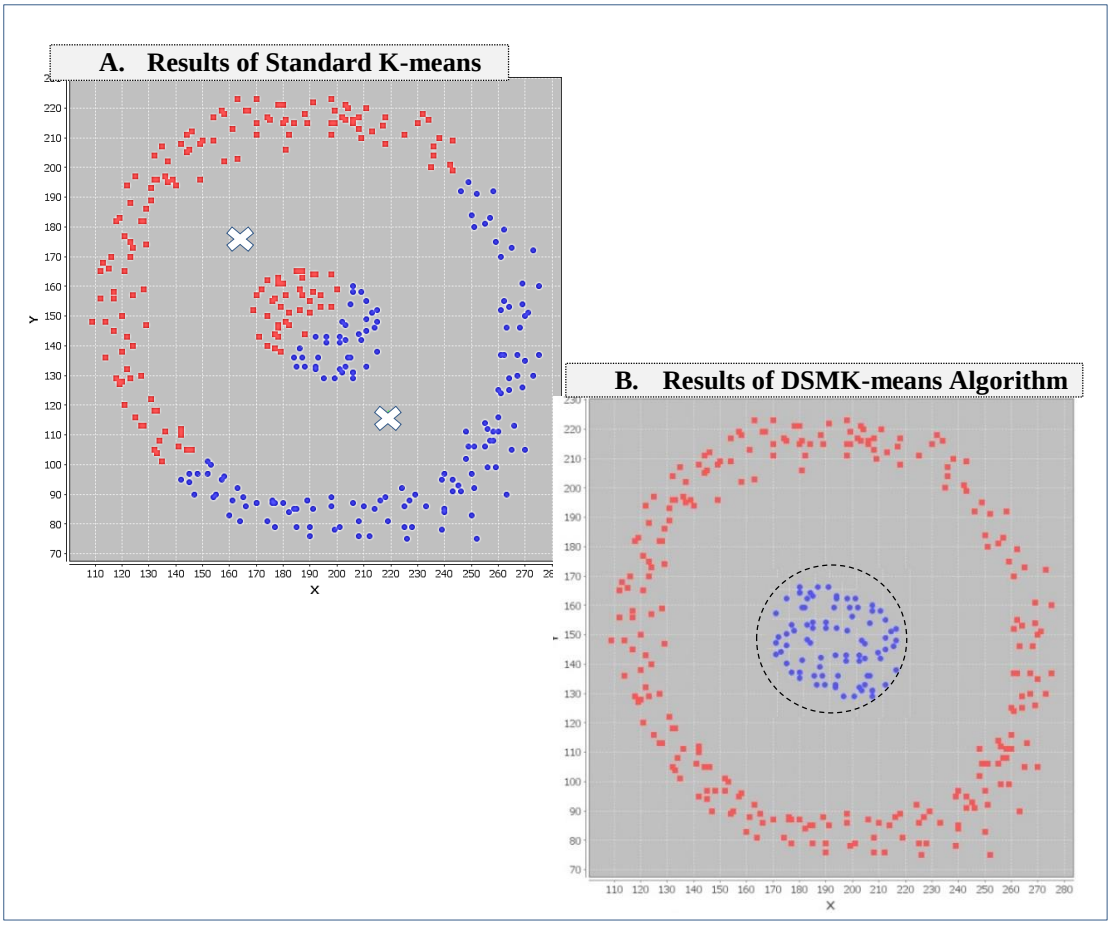

Figure 3.14: Low accurate result obtained with K-means V.S high accurate result obtained with DSMK-means algorithm.

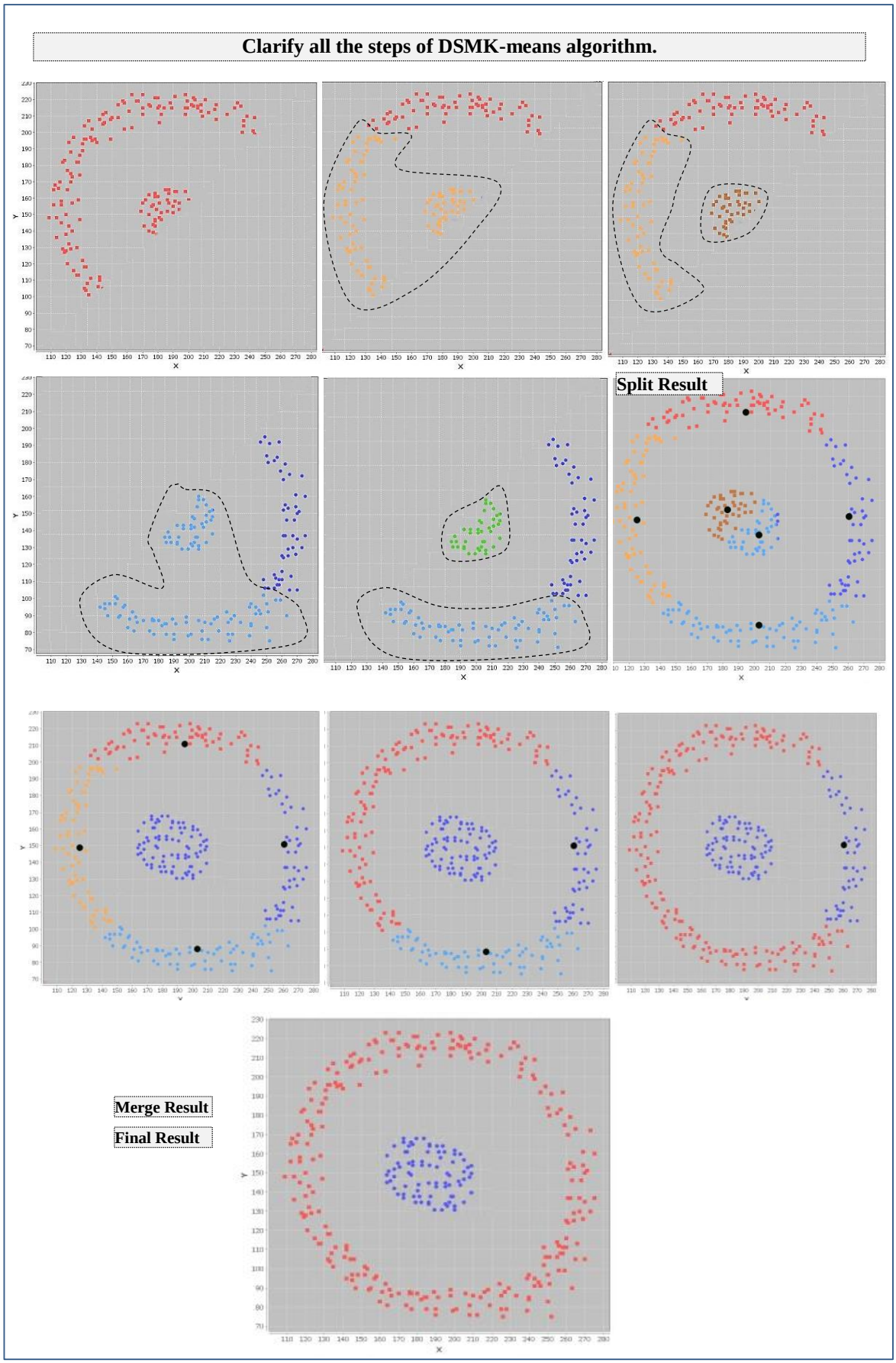

Figure 3.15: Proposed Split and merge Method steps.

❖ **Case two - Anti-Noise proposed Method:**

In standard K-means clustering, when applied on datasets containing noise objects , the results are -most of the time- of low accuracy. As the standard K-means includes all noise objects in the calculations, the end result will lack accuracy, in addition, standard K-means will either merge some true clusters into larger clusters or -in some cases- identify groups of noise points as clusters.

The researcher has developed a way to decrease the effect of noise objects on the end results through observations during lots of experiments applied on different datasets some of which were explained in the previous section. The researcher concluded that -most of the time- the noise points were in non-density unit as well as most of the points far from the centroids even when the K-means results are highly accurate. Based on that conclusion, the researcher build the Anti-noise method which is mainly about neglecting points far from the centroids in order to acquire high accuracy results.

Anti-noise method starts with calculating distances between each point in a cluster and its centroid where the distance are listed in an ascending order. Starting with the farthest points -which has the largest distance-, Anti-noise checks if that point is located in density unit or not. If it was located in non-density unit, then it is temporarily neglected and the next farthest point is check. This process goes on until a point that is located in a density unit is found or all the points are checked. In the case of finding no points in density unit, then the whole cluster is neglected, and the next cluster is checked in the same manner.

After checking all clusters within the dataset, standard K-means is applied again on the dataset without the neglected points. The results of such run will have higher accuracy than those when including the neglected points and the resulting centroids will be very close to the true centroids. Afterwards, each of the neglected points is assigned to the cluster with nearest centroid.

➢ **Expressing Anti-Noise Method by Example:**

From the experiments this method is important and completes the “split and merge” method as it applied when all cluster centroids are located in density units.

Anti-noise method steps are shown in Figure 3.16, where Figure 3.16(A) is showing the results of standard K-means algorithm which easy to notice how bad the results are. In this case, the split and merge method can not be applied as all the centroids are in density units. Figure 3.16(B) shows the result of K-means algorithm on the Document_Sim

dataset without neglecting noisy points, and by comparing Figure 3.16(A) and Figure 3.16| (B) you can notice which points are neglected. When analyzing standard K-means results without including the neglected noise points, it is noticed that Anti-noise method was able to locate centroids almost exactly as the true centroids. Figure 3.16(C) shows the final results of DSMK-means algorithm, the algorithm relocates the neglected points and assigns each one to the closest clusters' centroids which lead to high accuracy of algorithm and makes it able to cluster datasets with different complex shapes or those who have noise points.

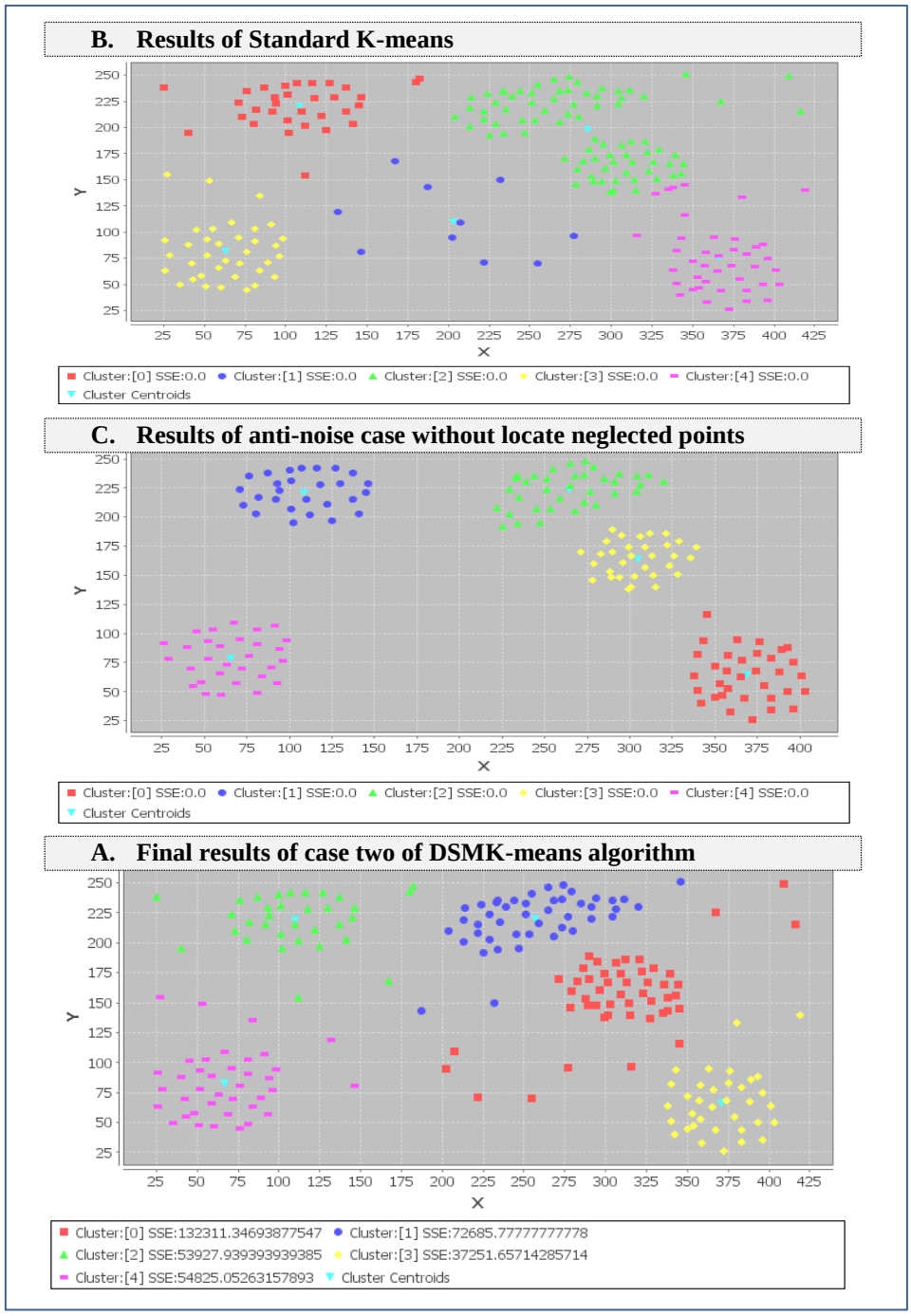

Figure 3.16: Proposed anti-noise method steps on Document_Sim dataset.

3.2.3 DSMK-means Algorithm Pseudo-Code

Suppose that we are going to partition $X=\{x_1,x_2,...,x_n\}$ which is a dataset with n number of objects, and k is an input parameter equal to number of clusters required.

Algorithm 3.2 : DSMK-means

Purpose: Clustering dataset

Input: $X=\{x_1,x_2,\ldots,x_j\}$ (set of entities to be clustered)
K (number of cluster)
MaxIters (Limit of iterations)

Output: $C=\{C_1,C_2,\ldots,C_K\}$ (set of cluster centroids)
L= (set of cluster labels of X)

Procedure

1. **RUN** standard K-means algorithm
2. **COMPUTE** sum square error "SSE" for each cluster.
3. **COMPUTE** ε and MinPts value for cluster with minimum "SSE" value.
   - → ***Eps or ε*** *, the radius that delimitate the neighbourhood area of a point (Eps-neighbourhood)*
   - → ***MinPts****, the minimum number of points that must exist in the Eps-neighbourhood.*
4. **FOREACH** cluster
5. Create list of clusters with centroids Ci in non-density units.
6. **IF** *number of point's within Eps-neighborhood contains < MinPts (centroid in density unit).*
7. **THEN** *add cluster to list Ci*
8. **ENDFOR EACH**
9. **CASE** METHOD **OF**
10. **CASE-ONE "*If one or more centroids Ci is not in density unit*":**
    (Spit and Merge started)
    11. **Declare** count=0 represent number of splitting operation

    SPLIT PROCESS

    12. **For** all Clusters C*i* List
    13. **IF** *centroid Ci is in non-density unit*
    14. Split C*i* cluster into two clusters with standard K-means (K=2).

15. DELETE C*i* cluster and ADD split clusters to List
16. Increase count by 1

17. ENDIF

18. ENDFOR

MERGE PROCESS

19. WHILE count != 0

20. Calculate centroids distance matrix
21. **FOR** each item in distance matrix
22. Find the tow nearest clusters centroids from all dataset clusters using distance matrix
23. Find the two closest points from the two closest clusters using single linkage.
24. **IF** (distance between two nearest points is less than or equals to ε) **THEN** Merge those two clusters.
25. **ELSE,** Find middle point
26. **IF (**middle point between two nearest points from two closest clusters "Single Linkage" is in density unit) **THEN**
27. Merge those two clusters.
28. Decrease count by 1
29. **ELSE,**
30. Go To step 22
31. **ENDIF**
32. **ENDFOR**
33. **IF** no clusters are merged THEN
34. Merge tow nearest clusters' centroids
35. Decrease count by 1
36. **ENDIF**

37. ENDWHILE

38. CASE-TWO *" If all centroids are in density units"*:

39. **FOR** each cluster Ci
40. **FOR** each point Pn
41. Compute distance between the centroid and Pn where n is the number of points in a cluster Ci

42. **ENDFOR**
43. Sort the distances in ascending order in each Ci
44. **ENDFOR**
45. **FOR** all centroids
46. **WHILE** the farthest point from centroid Ci is not in density unit
47. Neglect the point and considered as noise
48. **ENDFOR**
49. **RUN** standard K-means algorithm without the neglected points "noise"
50. Depending on the K-means cluster results, the neglected points are assigned to the closest cluster.
51. **ENDCASE**
52. **End ALGORITHM**

3.2.4 Advantages and Limitations of DSMK -Means Algorithm

- **Advantages:**
    1. The algorithm can handle large numbers of datasets as it solves two different problems in standard K-means (sensitivity to noise, complex shapes).
    2. The algorithm has combined the characteristics of partition clustering and density clustering concepts.
    3. The algorithm is not difficult to implement.
    4. The algorithm does not require any additional parameters more than the standard K-means algorithm.
    5. The algorithm is less sensitive to noise and outlier.
    6. Algorithm got better accuracy when datasets containing clusters with complex shapes and sizes.
    7. Algorithm able to cluster non-linearly separable data.

- **Limitations:**
    1. Algorithm did not reduce the number of parameters needed.
    2. Algorithm increases the computational complexity.
    3. In some rare cases, algorithm had bad results as the standard K-means.

The next Figures (3.17 – 3.18) exhibits the flow chart of the DSMK-means algorithm:

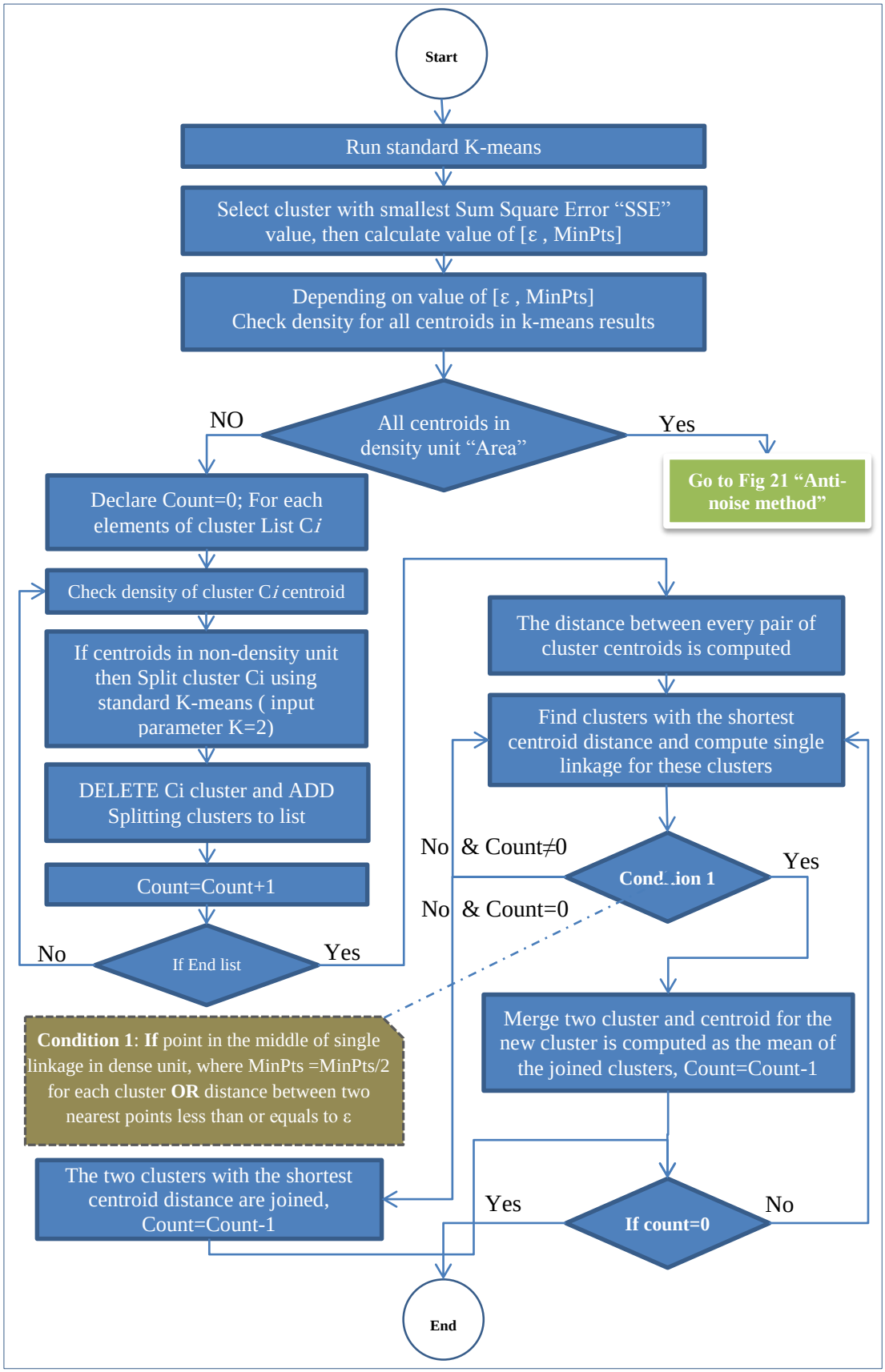

Figure 3.17: Flowchart of DSMK-means "Split and Merge" algorithm.

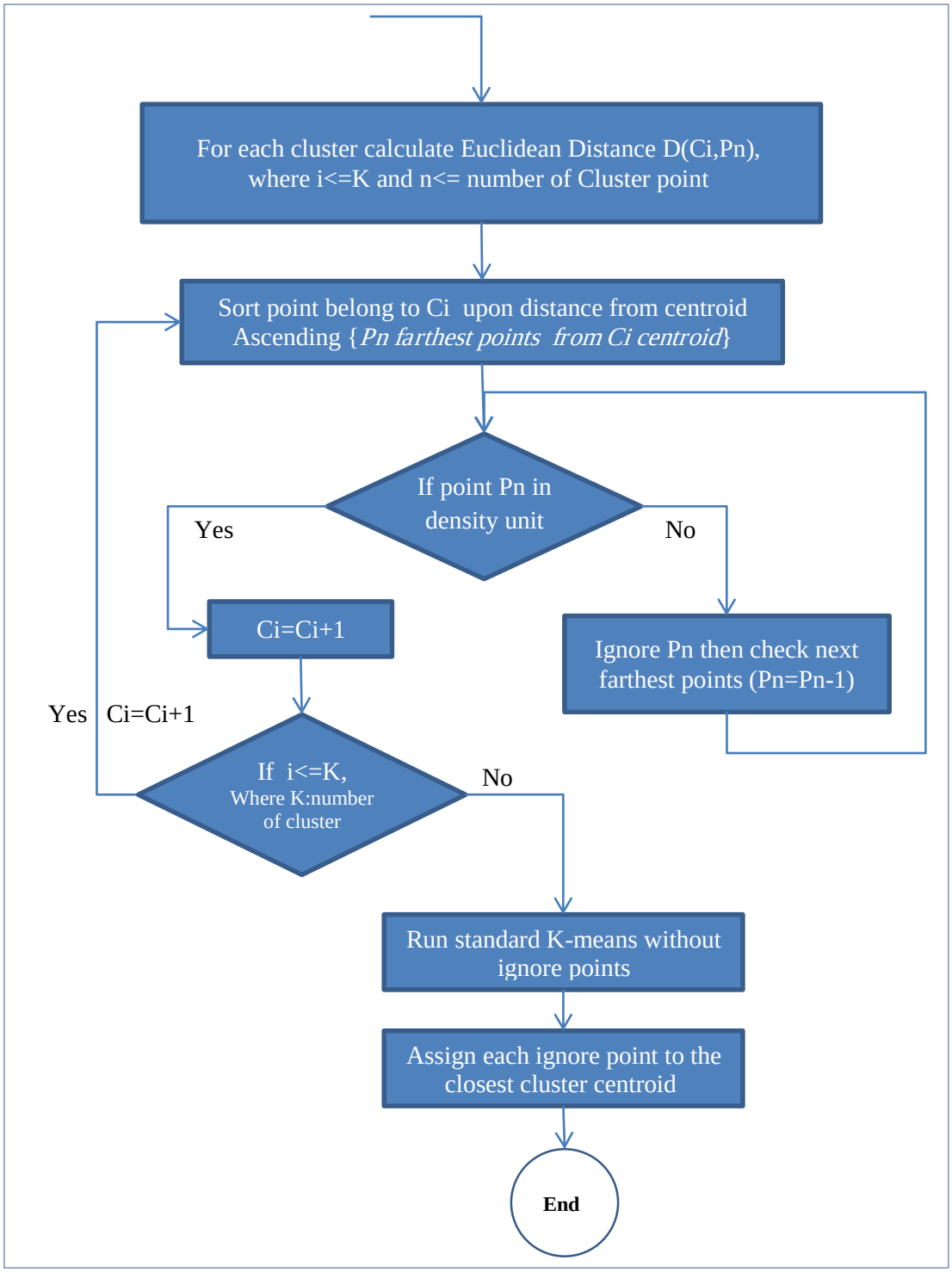

Figure 3.18: Flowchart of DSMK-means "Anti-noise" algorithm.

Chapter 4

4. Experiments Results

Chapter 4

4. Experimental Results

Description of the datasets used in experiments and the measurement techniques in addition to measuring the accuracy of the proposed algorithms' results to ensure their ability in delivering better results than other algorithms

5.1 Datasets Specifications

This section describes and identifies the specifications of datasets used in the all experiments on the proposed algorithms. The datasets varied between real world and artificial datasets.

5.1.1 Artificial Datasets

The Artificial datasets used in the experiments are:

- **The Ruspini dataset:**

Ruspini dataset [39], is a collection of 75 points, arranged in 4 groups, in the Euclidean plane. It is widely used to illustrate the effectiveness of clustering methods especially the effect of selecting the initial centroids.

- **The Rfivec dataset:**

Artificial dataset generated by the researcher with two dimensions, this dataset is designed in a way that is sensitive to centroid initialization. This dataset contains 135 points distributed into five true clusters. Values of the generated artificial dataset are used to assess the level of the algorithm accuracy and ability to identify true clusters.

- **Rnoisy dataset:**

Artificially polluted datasets with noise generated by the researcher with two dimensions, this dataset designed in a way to contain a lot of noise and outliers. This dataset consists of 188 points distributed in six true clusters. Values of the generated artificial dataset are

used to assess the level of K-means algorithm accuracy and ability to identify true clusters.

- **Ground_Separation dataset:**

Dataset contains six different complex shapes and sizes generated by the researcher with two dimensions. The dataset consists of 479 points distributed into six true clusters. This dataset was designed to be “hard” because of different clusters’ shapes. It is designed to measure K-means ability to identify clusters with complex shapes.

- **Separation_2Circle dataset :**

Dataset generated by the researcher with two different complex shapes and sizes with two dimensions. The dataset consists of 337 points in two true clusters. This dataset is designed to be “hard” because of different clusters’ shapes. It is designed to measure K-means ability to identify clusters with complex shapes.

- **Document_Sim dataset:**

Document_Sim dataset [40] generated so that many noises are scattered. The dataset consists of 200 points in five true clusters. The dataset is designed to be "hard". i.e. there is a large number of outliers and noise are scattered between five true clusters. It is designed to measure K-means ability to identify true clusters in noisy datasets.

- **Aggregation dataset:**

Aggregation dataset [41] consists of the seven perceptually distinct clusters with different shapes. The dataset consists of 788 points distributed in seven true clusters. The dataset designed to be "hard" in order to measure K-means ability to identify clusters with complex shapes.

5.1.2 Real Datasets

All datasets used in the following experiments and more can be found in UCI Machine Learning Repository [42] which is a collection of databases, domain theories, and data generators used by the machine learning community for the empirical analysis of machine learning algorithms.

❖ **Iris Dataset:**

This is perhaps the best-known database to be found in the pattern recognition and clustering literature. The Iris flower dataset or Fisher's Iris dataset is a multivariate dataset introduced by Sir Ronald Fisher (1936) as an example of discriminant analysis. It is sometimes called Anderson's Iris dataset because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species, which are shown in Figure 4.1 [43]. The dataset contains three classes of 50 instances each, where each class refers to a type of Iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. Table 5.1 illustrates the specifications of the dataset.

Table 5. 1 Iris dataset specifications

Dataset Characteristics:	Multivariate	**Number of Instances:**	150
Attribute Characteristics:	Real	**Number of Attributes:**	4
Associated Tasks:	Classification, Clustering	**Missing Values?**	NO
Area	Life		
Attribute Information:	**1.**	Sepal length in cm	
	2.	Sepal width in cm	
	3.	Petal length in cm	
	4.	Petal width in cm	
	5.	**GROUPS**	→ Iris Setosa → Iris Versicolour → Iris Virginica

Figure 4.1: three related species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor).

❖ **Libras Movement Dataset:**

The dataset contains 15 classes of 24 instances each, where each class references to a hand movement type in LIBRAS as exhibited in Figure 4.2. In the video pre-processing, time normalization is carried out selecting 45 frames from each video, in according to a uniform distribution. In each frame, the centroid pixels of the segmented objects (the hand) are found, which compose the discrete version of the curve F with 45 points. All curves are normalized in the unitary space. In order to prepare these movements to be analyzed by algorithms, a mapping operation is carried out, that is, each curve F is mapped in a representation with 90 features, with representing the coordinates of movement. Some sub-datasets are offered in order to support comparisons of results. Table 5.2 illustrates the dataset specifications.

Table 5. 2 Libras Movement dataset specification

Dataset Characteristics:	Multivariate, Sequential	**Number of Instances:**	360
Attribute Characteristics:	Real	**Number of Attributes:**	91
Associated Tasks:	Classification, Clustering	**Missing Values?**	NO
Attribute Information:	**1-91**	90 numeric (double) and 1 for the class (integer)	

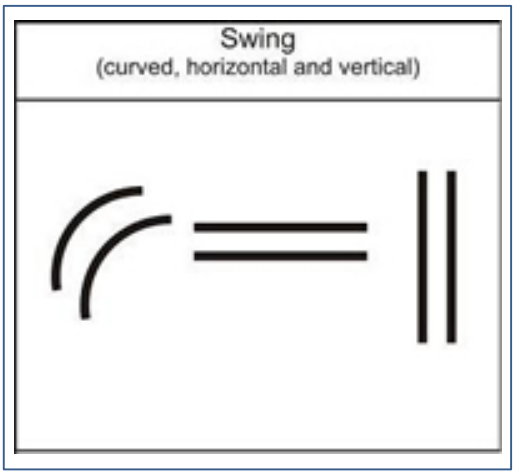

Figure 4.2: Swing (Curved, horizontal, and vertical).

- **Weblogs dataset:**

Weblogs dataset [40] is real with two dimensions; it is suitable to describe the performance of K-means when dealing with datasets containing clusters with different complex shapes and sizes. The datasets contain two metadata of weblog entries: number of visits and purchase. Figure 4.3 shows the dataset point distribution. The datasets had been gathered by crawling from the WWW. Table 5.3 illustrates the dataset specifications.

Table 5.3 Weblogs dataset specification

Dataset Characteristics:	Multivariate	**Number of Instances:**	192
Attribute Characteristics:	Integer	**Number of Attributes:**	3
Associated Tasks:	Classification, Clustering	**Missing Values?**	NO
Attribute Information:	**1.**	Number of visits	
	2.	Number of purchase	

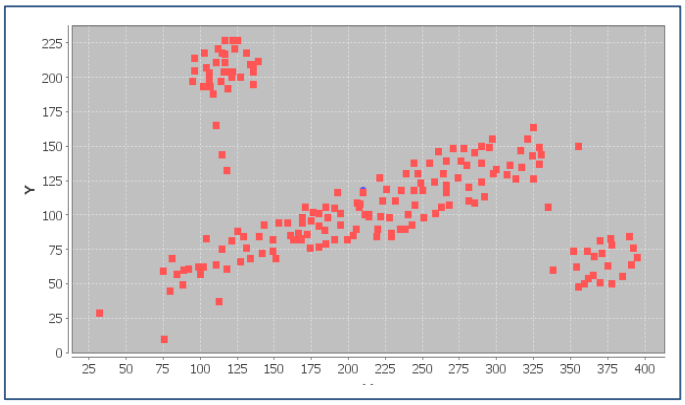

Figure 4.3: illustration of Weblogs dataset.

- **Image_Extraction dataset:**

This dataset is simplified extraction of local image features. It takes image data as input and returns a dataset with feature vectors computed from image blocks on a regular grid as exhibited in Figure 4.4. The dataset consists of samples from each of two species of image. Table 5.4 illustrates the dataset specifications.

Table 5.4 Image_Extraction dataset specification

Dataset Characteristics:	Multivariate	**Number of Instances:**	200
Attribute Characteristics:	Integer	**Number of Attributes:**	2
Associated Tasks:	Clustering	**Missing Values?**	NO
Attribute Information:	**1.**	Image ID	
	2.	Color	

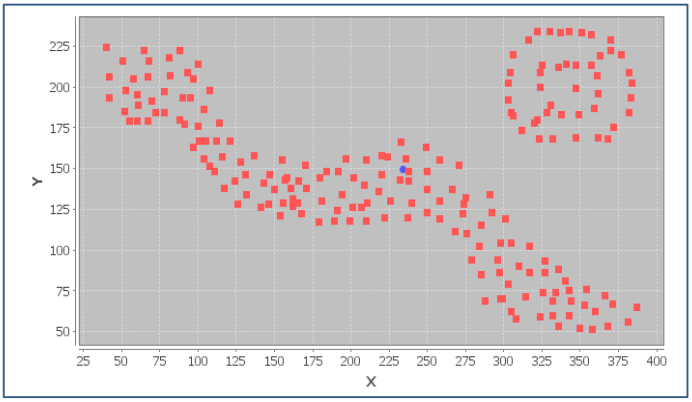

Figure 4.4: Illustration of Image_Extraction dataset

❖ Mammographic Mass Dataset:

The most effective method for breast cancer screening available today is Mammography (illustrated in Figure 4.5). However, the low positive predictive value of breast biopsy resulting from mammogram interpretation leads to approximately 70% unnecessary biopsies with benign outcomes. To reduce the high number of unnecessary breast biopsies, several computer-aided diagnosis (CAD) systems have been proposed in the last years. These systems help physicians in their decision to perform a breast biopsy on a suspicious lesion seen in a mammogram or to perform a short term follow-up examination instead. This dataset can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. It contains a BI-RADS assessment, the patient's age and three BI-RADS attributes together with the ground truth (the severity field) for 516 benign and 445 malignant masses that have been identified on full field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006. Each instance has an associated BI-RADS assessment ranging from 1 (definitely benign)

to 5 (highly suggestive of malignancy) assigned in a double-review process by physicians. Assuming that all cases with BI-RADS assessments greater or equal a given value (varying from 1 to 5), are malignant and the other cases benign, sensitivities and associated specificities can be calculated. These can be an indication of how well a CAD system performs compared to the radiologists. Table 5.5 illustrates the dataset specifications.

Table 5.5 Mammographic Mass dataset specification

Dataset Characteristics:	Multivariate	**Number of Instances:**	961
Attribute Characteristics:	Integer	**Number of Attributes:**	6
Associated Tasks:	Classification, Clustering	**Missing Values?**	YES
Area	Life		
Attribute Information: "*Attributes in total (1 goal field, 1 non-predictive, 4 predictive attributes)* "	**1.**	BI-RADS assessment: 1 to 5 (ordinal, non-predictive!)	
	2.	Age: patient's age in years (integer)	
	3.	Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)	
	4.	Margin: mass margin: circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5 (nominal)	
	5.	Density: mass density high=1 iso=2 low=3 fat-containing=4 (ordinal)	
	6.	Severity: benign=0 or malignant=1 (binominal, goal field!)	
Missing Attribute Values:	**1.**	BI-RADS assessment: 2	
	2.	Age: 5	
	3.	Shape: 31	
	4.	Margin: 48	
	5.	Density: 76	
	6.	Severity: 0	

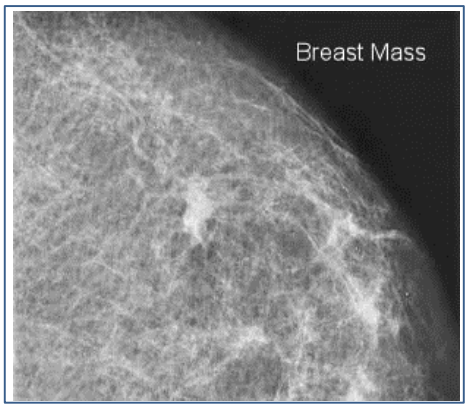

Figure 4.5: illustration of Mammographic Mass.

Table 5.6 presents a summary of artificial datasets used in this thesis while Table 5.7 presents a summary of real datasets. The details of each dataset are described.

Table 5.6 summary of all artificial datasets information

Datasets Name	clusters	Point number	type	dimension
Ruspini	4	75	Real	2
Rfivec	5	135	Real	2
Rnoisy	6	188	Integer	2
Ground_Seperation	6	479	Integer	2
Separation_2Circle	2	337	Integer	2
Document_Sim	5	200	Integer	2
Aggregation	7	788	Real	2

Table 5.7 summary of all Real datasets information

Datasets Name	clusters	Point number	type	dimension
IRIS	3	150	Real	3
Libras Movement	15	360	Real	90
Web Log	3	192	Real	3
Image_Extraction	2	200	Real	2
Mammographics_Mass	2	961	Real	6

5.2 Cluster Validity Measures and Experiments Environment

Evaluation of clustering results sometimes is referred to as cluster validation. There have been several suggestions for a measure of quality of clustering algorithms. Such a measure can be used to compare how well different clustering algorithms perform on a set of data. These measures are usually tied to the type of criterion being considered in assessing the quality of a clustering algorithm [44].

5.2.1 Measuring clustering validity

- **External validity**:

In external validity, clustering results are evaluated based on already clustered data such as known class labels and external benchmarks. Such benchmarks consist of a set of pre-classified items, and these sets are often created by human (experts). Thus, the benchmark sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined benchmark classes. In summary, external evaluation measures similarity of clustering against known class labels.

- **Internal validity**:

When a clustering result is evaluated based on the data that was clustered itself, this is called internal validity. These methods usually assign the best score to the algorithm that produces clusters with high similarity within a cluster and low similarity between clusters. One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information retrieval applications. In summary, internal validity measure the goodness of a clustering without any external information just like Sum of Squared Error (SSE), Akaike Information Content score (AIC), The Bayesian Information Criterion (BIC), and Sum Of Average Pairwise Similarities (SAPS):.

Researcher evaluates the effectiveness of proposed clustering algorithms by the following internal evaluation measurement algorithms:

- **Sum of Square Errors (SSE):** [45] SSE is the simplest and most widely used criterion measure for clustering. For a given cluster; SSE is computed as follows: for each instance in the cluster; summing the square differences between each

attribute value and the corresponding one in the cluster centroid. These are summed up for each instance in the cluster and for all clusters. The formula for SSE for one cluster is:

$$\mathrm{SSE} = \sum_{i=1}^{n} (x_i - x_c) \qquad (4.1)$$

Where n is the number of observations xi is the value of the ith observation and xc is the mean of all the observations.

- **Akaike Information Content (AIC) score:** [46] [47] is a measure of the relative quality of a statistical model, for a given set of data. As such, AIC provides a means for model selection. AIC is founded on information entropy: it offers a relative estimate of the information lost when a given model is used to represent the process that actually generates the data. AIC deals with the trade-off between the complexity of the model and the goodness of fit of the model. AIC measures the log-likelihood of the model penalized by the number of parameters in the model. A clustering result with small K and small variance of each cluster will have a relatively low AIC score, which means the clustering result is good. In the general case, the formula for AIC is:

$$\mathrm{AIC} = 2k - 2\ln(L) \qquad (4.2)$$

Where k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model.

- **The Bayesian Information Criterion (BIC):** [48] BIC proposed by Schwarz (1978) is a popular method for model selection. BIC evaluates candidate models with different number of basic functions, and the optimal number is chosen from the best model in terms of BIC score. The formula for the BIC is:

$$\mathrm{BIC} = 2 * \ln(k) + k * \ln(n) \qquad (4.3)$$

Where n is equal to sample size, k is the number of parameters in the statistical model, and L is the maximized value of the likelihood function for the estimated model.

- **Sum of Average Pairwise Similarities (SAPS):** [49] The average pairwise difference within a population can be calculated as the sum of the pairwise differences divided by the number of pairs.

5.2.2 Experiments Environments specification

The experiments are performed on a laptop with {*Intel core i5 i5-3210m / 2.5 ghz (3.1 ghz)(dual-core)*} processor and 6 gigabyte memory running Microsoft windows 8 professional 64-bit edition operating systems. Java programing language (*Java™ Platform, Standard Edition Development Kit (JDK™)*) is used to code the algorithms.

5.3 Performance Evaluation of DMIK-Means Algorithm

To test performance of "DIMK-means" the researcher used internal validity with selected datasets, then, the tests results were compared with standard k-mean algorithm.

5.3.1 Datasets selection

The performance evaluation of DIMK-means is applied on five different artificial and real-world datasets (Ruspini, Rfivec, IRIS, and Libras Movement) using popular evaluation methods including: Sum of Square Errors (SSE), Akaike Information Content (AIC) and The Bayesian Information Criterion (BIC), which were described in the previous sub section. The results of such evaluation are compared with standard K-means algorithm with random initialization method in order to identify the differences.

Table 5.8 shows the comparison of standard K-means performance results using the random initialization and DIMK-means algorithm, which was applied on the artificial datasets described in 5.1.1 subsection.

From the table 5.8 its observed that DIMK-means algorithm scored smaller values for each type of performance measures (SSE, AIC, or BIC) than the results of standard K-means on both artificial datasets Ruspini, and Rfivec. Which means that DIMK-means get high accurate results compared with standard k-means algorithm. The researcher observes the difference between the values of the worst and the best case reached by the standard K-means algorithm, which initialized with the random method, is very high, while in DIMK-means, the gap between the worst and best case is kept to minimum. This proves that DIMK-means is more stable than standard K-means and has better results working with artificial datasets.

Table 5.8 The algorithms mean results of artificial datasets over 30 runs (K is an input parameter obtained from user, which represent the number of clusters).

Dataset	algorithm	K	SSE	AIC	BIC
Ruspini	K-means	**4**	69878.823153	2989.2333511	2989.108201
	DMIK-means		25712.90025	2948.02156	2947.895195
Rfivec	K-means	5	630730.32154	6583.4986813	6583.628358
	DMIK-means		473906.42563	6567.4043256	6567.535321

Table 5.9 also shows the comparison of initial cluster centroids computed using DIMK-means and standard K-means algorithms, which were applied on the real datasets described before.

From the table 5.9 its observed that DIMK-means algorithm scored smaller values for each type of performance measures (SSE, AIC, or BIC) than the results of standard K-means on both real world IRIS, and Libras Movement datasets. This mean that DIMK-means get high accurate results compared with standard k-means algorithm when working on both type of datasets real world and artificial datasets. However, the results of DIMK-means are pretty good since the difference between the values of the worst and the best case for Iris, and Libras Movement datasets, is kept to minimum, while the value reached by the standard K-means algorithm, is very high. This proves that DIMK-means is more stable than standard K-means and has better results working with real world datasets.

Table 5. 9 The algorithms mean results of real datasets over 30 runs (K is an input parameter obtained from user, which represent the number of clusters)

Dataset	algorithm	K	SSE	AIC	BIC
IRIS	K-means	**3**	223.3735747	9926.572673	9926.748764
	DMIK-means		171.1536034	9903.996737	9904.172828
Libras Movement	K-means	15	678.0630313	2109578.518	2109577.961
	DMIK-means		649.1821375	2087765.609	2087765.052

Experiments show that our proposed method is more efficient and stable than standard K-means algorithms. That DIMK-means algorithm scored smaller values for each type of performance measures (SSE, AIC, or BIC) than the results of standard K-means. In addition, usually DIMK-means leads to SSE values close to or less than the minimum SSE values obtained from standard K-means. This proves that the proposed method is more stable than the random method and has better results confirming the need for a stable initialization method.

To prove the efficiency of DIMK-means, the graph of the artificial datasets visually illustrates a comparison between the results of the standard K-means and DIMK-means algorithms as Figures 4.6 and 4.7 show the results of running the two algorithms with Ruspini dataset, which consists of four clusters.

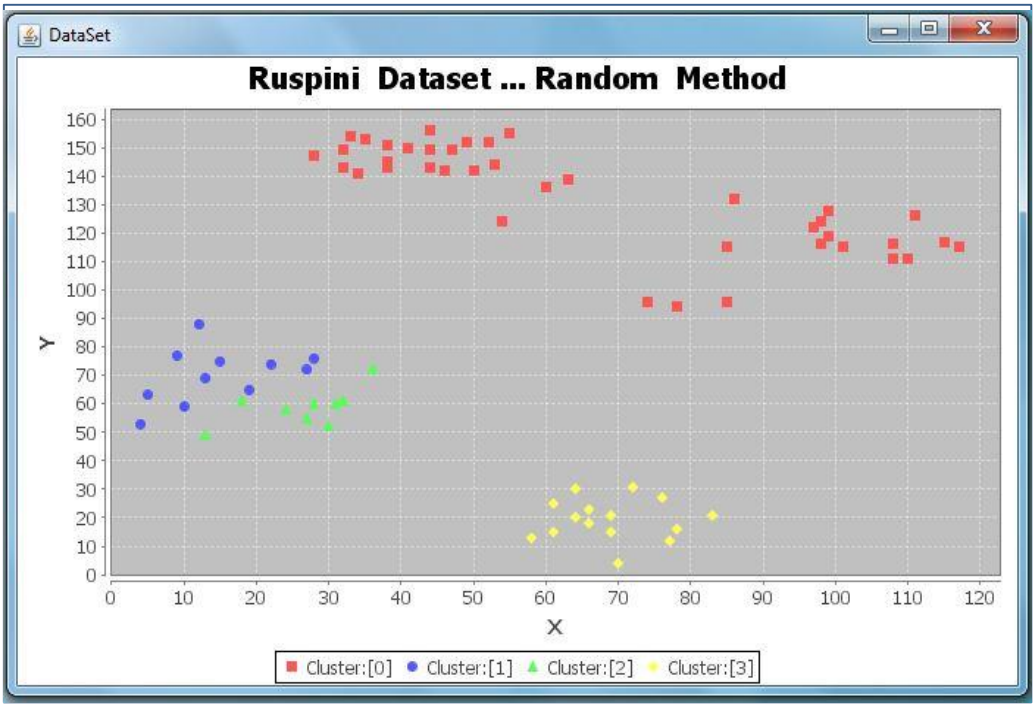

Figure 4.6: Results of running the standard K-means with k=4 and using 4 different starting points, each randomly chosen from the dataset

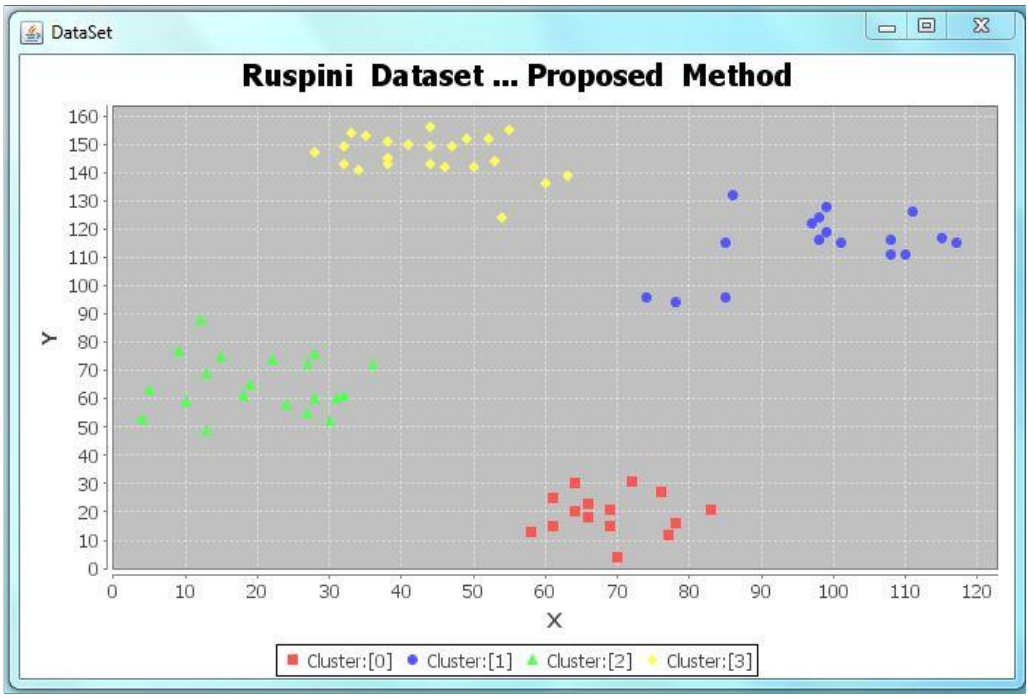

Figure 4.7: Results of running DIMK-means with k=4 and using 4 different starting points, each chosen with the proposed initialization method.

In Figure 4.6 and 4.7, each identified cluster was demonstrated using a different plotting character and color. Note the widely divergent results where it is observed that the random method for initialization in standard K-means gets inefficient results as it merges 2 clusters together (which are plotted as red square in Figure 4.6) and split one of the true clusters to two different clusters (plotted with blue circles and green dots in Figure 4.6). While the proposed DIMK-means algorithm is more efficient and accurate in identifying each cluster very close to the true ones.

Figures 4.8 and 4.9 show the results of running the standard K-means and DIMK-means algorithms respectively with Rfivec dataset which consists of five clusters.

like the previous figures, it is observed that the random method for initialization in standard K-means gets inefficient results as it merges 2 clusters together (which are plotted as blue circle in Figure 4.8), splits one of the true clusters to tow different clusters (plotted with red square and pink oblong in Figure 4.8) and merges subset of cluster with another one (plotted with yellow cube in Figure 4.8). While DIMK-means is more efficient and accurate in identifying each cluster very close to the true ones as shown in Figure 4.9.

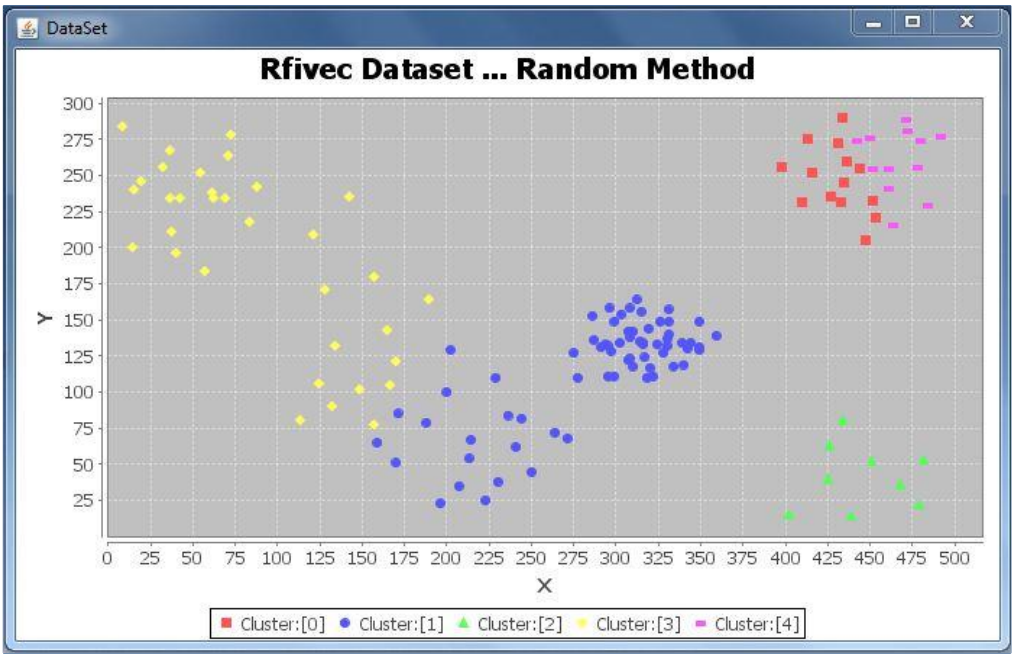

Figure 4.8: Results of running the standard K-means with k=5 and using 5 different starting points, each randomly chosen from the dataset.

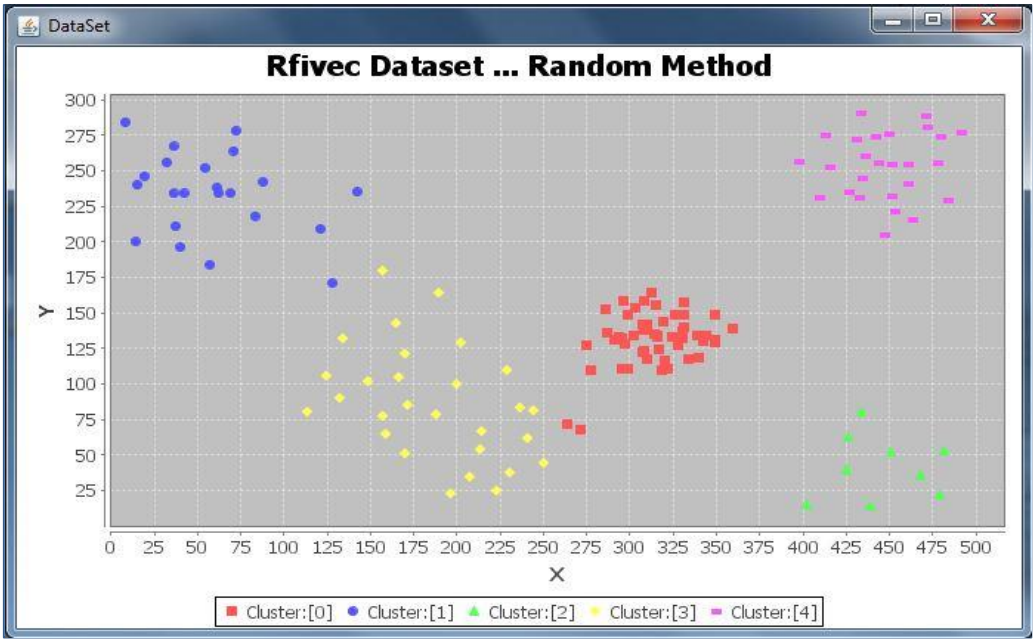

Figure 4.9: Results of running DIMK-means with k=5 and using 5 different starting points, each chosen using the proposed initialization method.

5.4 Performance Evaluation of DSMK-Means Algorithm

To test the performance of "DSMK-means" algorithm, the researcher here introduces the datasets used in the test and reviews the results of the experiments, comparing the results with standard K-means algorithm and "BNAK-Divide-and-Merge Clustering Algorithm (BNAKDAM) [36]".

5.4.1 Datasets Selection

The performance evaluation of DSMK-means algorithm is applied on nine different artificial and real-world datasets (Ground_Separation, Separation_2Circle, Rnoisy, Aggregation, Document_Sim, Weblogs, Image_Extraction, and Iris). Furthermore, the performance of DSMK-means algorithm is evaluated using popular internal clustering validity indices, which employed to evaluate the clustering results, such indices include: Sum of Square Errors (SSE), Akaike Information Content (AIC), The Bayesian Information Criterion (BIC), and Sum of Average Pairwise Similarities (SAPS); which were described in the previous section. The results of such evaluation are compared with standard K-means and BNAKDAM algorithms in order to identify the differences.

Table 5.10 and Table 5.11 show the comparison of the three clustering algorithms: Standard K-means, BNAKDAM, and proposed DSMK-means algorithm.

Table 5.10 shows the comparison applied on the artificial datasets described in 5.1.1 subsection, while Table 5.11 shows the comparison applied on the real datasets described before in 5.1.2 subsection.

Table 5. 10 Clustering algorithms mean results of artificial datasets over 50 runs (K is an input parameter obtained from user, which represent clusters number).

Dataset	algorithm	K	SSE	AIC	BIC	SAPS
Ground_Separation	K-means	**6**	**2855774.813**	**21565.0806**	**21565.74618**	**447.3896701**
	BNAKDAM		**4845704.853**	**21528.5757**	**21529.24129**	**439.9237214**
	DSMK-means		**6159061.753**	**21494.59273**	**21495.25831**	**435.4722115**
Separation_2Circle	K-means	**2**	**1646936.594**	**14267.35041**	**14267.87804**	**333.0847394**
	BNAKDAM		**2154097.931**	**14254.68015**	**14255.20778**	**328.7270521**
	DSMK-means		**2577578.554**	**14253.17946**	**14253.70709**	**325.1817196**
Document_Sim	K-means	**5**	**1306929.15**	**14744.83774**	**14745.38305**	**342.5543868**
	BNAKDAM		**858283.6622**	**14586.10662**	**14586.65193**	**341.2153644**
	DSMK-means		**563265.0873**	**14480.64838**	**14481.19369**	**339.6457059**

Dataset	algorithm	K	SSE	AIC	BIC	SAPS
Rnoisy	K-means	**6**	**1486294.884**	**21177.29713**	**21177.57129**	**183.0293608**
	BNAKDAM		**865616.0755**	**19627.6786**	**19727.92254**	**172.2823015**
	DSMK-means		**601464.5215**	**19143.84816**	**19144.07983**	**167.9597529**
Aggregation	K-means	**7**	**23875.26004**	**49524.31143**	**49525.20795**	**771.7122547**
	BNAKDAM		**24395.60533**	**49454.00687**	**49454.90339**	**772.9397117**
	DSMK-means		**25387.49156**	**49358.04323**	**49358.93976**	**769.6763346**

It is observed from the experiment results on artificial datasets described in Table 5.10, that DSMK-means has the best results among the other two algorithms in AIC, BIC, and SAPS indices, while it did not have the best results with SSE index. The reason of the SSE high score in DSMK-means algorithm depends on that the shape of cluster, as SSE sums the square differences between each attribute value and the corresponding one in the cluster centroid. In another example, SSE results for the Separation_2Circle dataset using standard K-means algorithm as shown in Figure 4.10.(A) were lower than the results using DSMK-means as in Figure 4.10.(F,E). It is known that the lower SSE score is the better, but in this case it is visually clear that Figure 4.10.(A) -which obtained lower SSE value- is very bad clustering result compared to the resulting cluster of DSMK-means. This observation indicates the SSE score can not be used to judge the clustering accuracy in cases of complex shapes. While the other indices give more accurate indication for the best clustering results.

Table 5. 11 Clustering algorithms mean results of real datasets over 30 runs (K is an input parameter obtained from user, which represent the number of clusters)

Dataset	algorithm	K	SSE	AIC	BIC	SAPS
Weblogs	K-means	3	**1001992.83**	**8670.09057**	**8670.369323**	**187.3771721**
	BNAKDAM		**1326821.065**	**8747.432511**	**8747.711265**	**187.8597177**
	DSMK-means		**1552251.178**	**8631.13863**	**8600.417384**	**186.1648071**
Image_Extraction	K-means	2	**1799087.147**	**9185.98039**	**9186.28142**	**191.6552485**
	BNAKDAM		**2991817.961**	**9305.285303**	**9305.586333**	**184.5466748**
	DSMK-means		**3634144.897**	**9069.956687**	**9070.257717**	**180.7335421**
IRIS	K-means	3	**223.37357471**	**9926.572673**	**9926.748764**	**149.5602358**
	BNAKDAM		**162.35621565**	**9912.365894**	**9950.464253**	**149.5591551**
	DSMK-means		**158.28685748**	**9899.236799**	**9899.4128911**	**149.5536856**

It is observed that DSMK-means algorithm scores smaller values for each type of clustering validity indices (SSE, SAPS, AIC, and BIC) where the sometimes, DSMK-means algorithm scores big values for clustering validity index (SSE). The value of measurement algorithm depends on the nature of algorithm formula and datasets clusters shapes, however DSMK-means could identify clusters with different complex shapes that may increase the result of SSE index while decrease the rest of indices results. Obviously, the clustering results of the DSMK-means clustering algorithm perform best compared to k-means and BNAKDAM clustering algorithms.

To prove the efficiency of DSMK-means algorithm, the graph of datasets is shown to make a comparison between the results of the standard K-means and DSMK-means algorithm. The BNAKDAM results were not shown here as they were similar to the graphs in Figure 4.10.(A,B,C).

- **Interpreting and compare results of DSMK-means And K-means algorithms with Separation_2Circle dataset:**

The Separation_2Circle is composed of two different clusters with different shapes. In Figure 4.10, the results show that the DSMK-means can detect both clusters with different shapes and sizes while the standard K-means cannot deal with this kind of dataset.

Each cluster identified by a different plotting character and color. It is observed that the standard K-means get inefficient results as it split the inner circle true cluster into two different groups as well as the outer circle; and merged each part of the inner circle with another part from the outer circle. It's observed that standard K-means always get the same results with this dataset which are very bad results (which are plotted as red square and blue circle in Figure 4.10.(A)). While the proposed DSMK-means algorithm gets more efficient and accurate results in identifying each cluster very close to true ones. It is worth mentioning that the results in Figure 4.10.(E and F) are the most common case in the results of the algorithm.

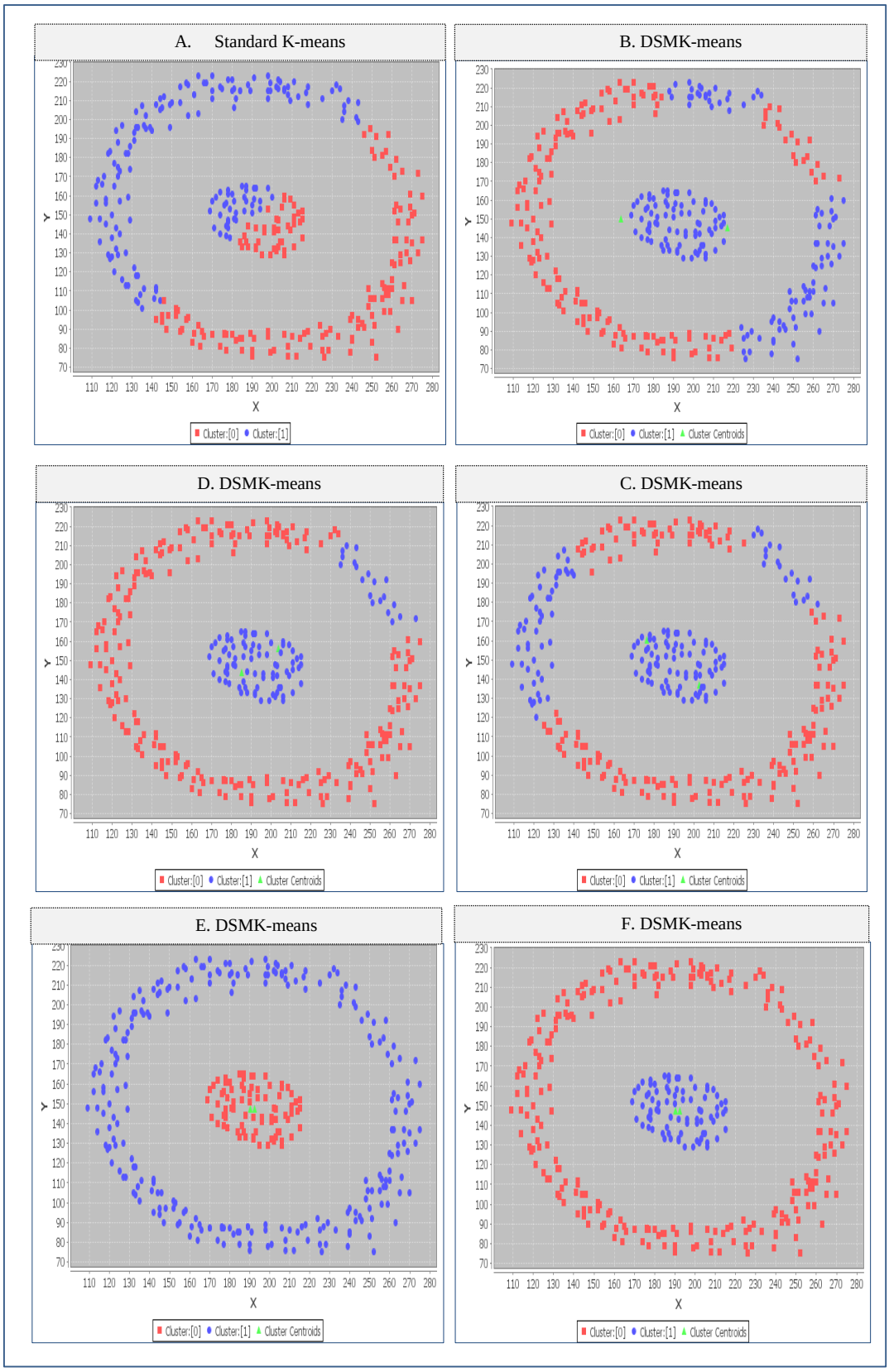

Figure 4.10: Results of running K-means and DSMK-means algorithms with K=2, on Separation_2Circle dataset.

❖ **Interpreting and compare results of DSMK-means And K-means algorithms with Image_Extraction dataset:**

Figure 4.11 shows the results of running standard K-means and DSMK-means algorithms with Image_Extraction dataset, which consists of two clusters.

It is observed that the standard K-means gets inefficient results as it splits the stripe-shaped cluster (which is plotted as red square in Figure 4.11.(B)) to two different groups (which are plotted as red squares and blue circles in Figure 4.11.(A)). Such results were repeated each time the standard K-means algorithm was applied to the same dataset. While the proposed DSMK-means algorithm gets more efficient and accurate results in identifying each cluster very close to the true ones.

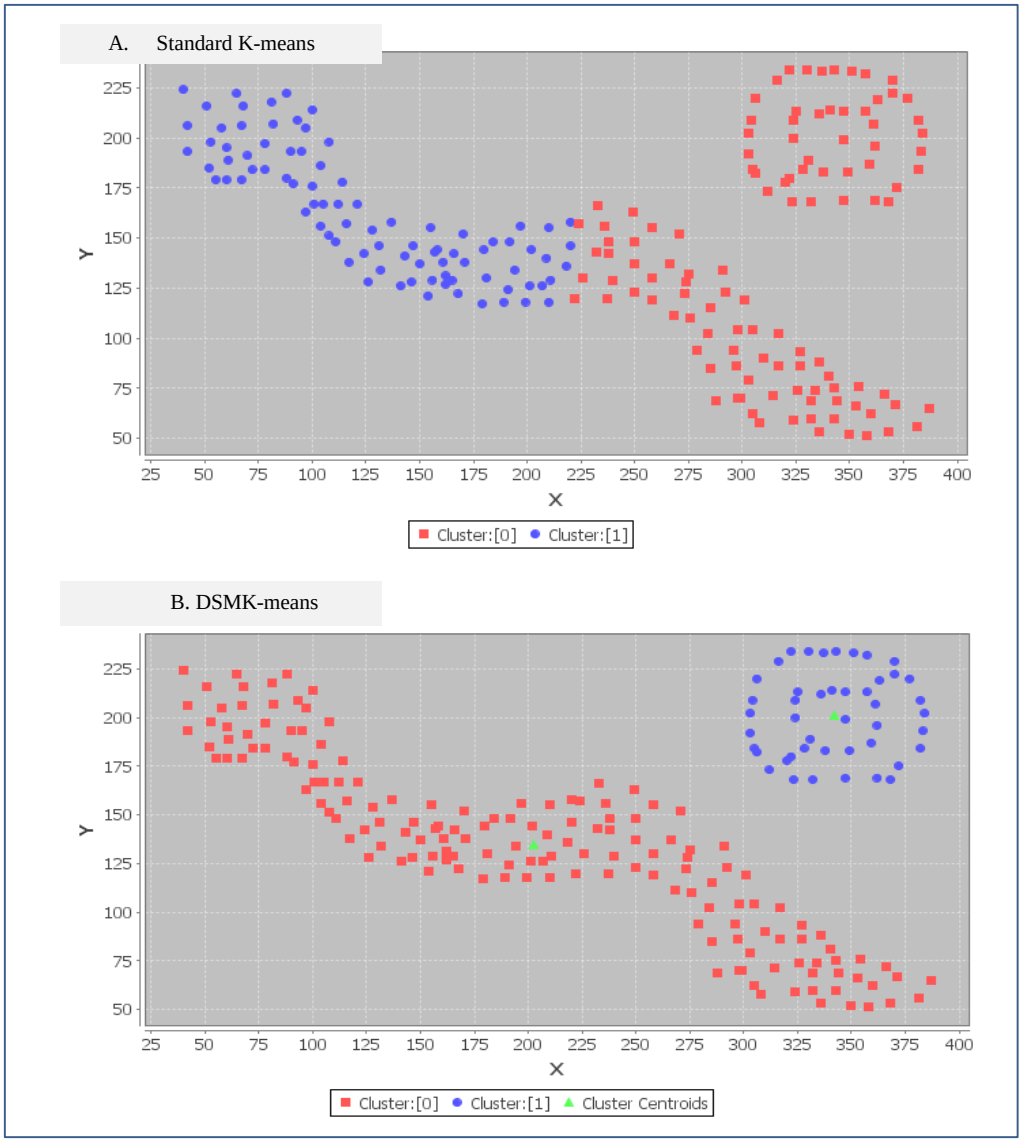

Figure 4.11: Results of running K-means and DSMK-means algorithms with K=2, on Image_Extraction dataset.

❖ **Interpreting and compare results of DSMK-means And K-means algorithms with Ground_Separation dataset:**

Figure 4.12 shows the results of running the K-means and DSMK-means algorithms with Ground_Separation dataset, which consists of 6 clusters. The shape of this dataset is one of the most complicated shapes to be tested on standard K-means, which is can not provide accurate clustering results that are close to the true clusters.

It is observed that the standard K-means get inefficient results as it split the ring-shaped cluster (which is plotted in Figure 4.12.(A)) into two different groups, one of them was identified as single cluster, while the other was merged with one of the circle-shaped clusters in the left-bottom corner.

On the other hand, the proposed DSMK-means algorithm get more efficient and accurate results in identifying each cluster very close to the true ones (which are plotted in Figure 4.12.(B)).

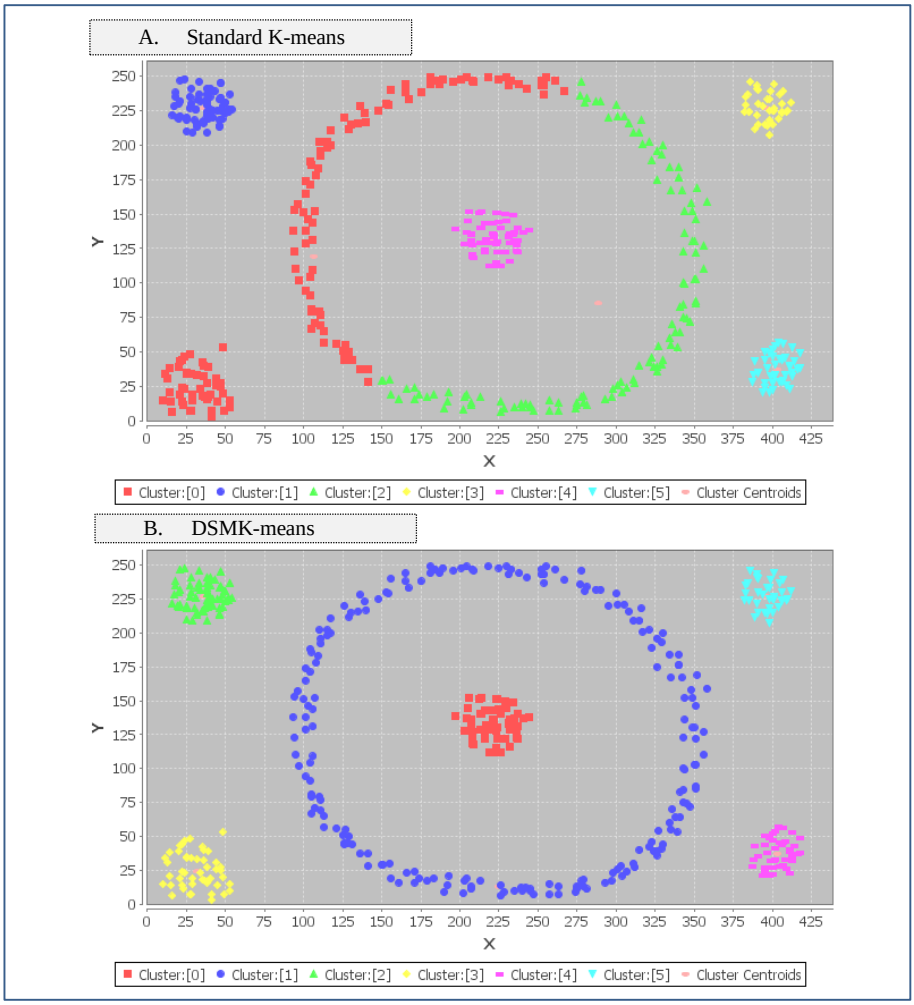

Figure 4.12: Results of running K-means and DSMK-means with k=6, with Ground_Separation dataset.

❖ **Interpreting and compare results of DSMK-means And K-means algorithms with Aggregation dataset:**

Figure 4.13 shows the results of running standard K-means and DSMK-means algorithms on Aggregation dataset that consists of seven clusters .

The Aggregation is composed of seven different clusters with different shapes, which are not well separated. In Figure 4.13.(A) the results show that standard K-means algorithm could not identify the truce clusters as in it split one of the true clusters (in Figure 4.13.(A) right image) into three different clusters.

DSMK-means algorithm could detect the true clusters with different shapes and sizes as shown in Figure 4.13.(B).

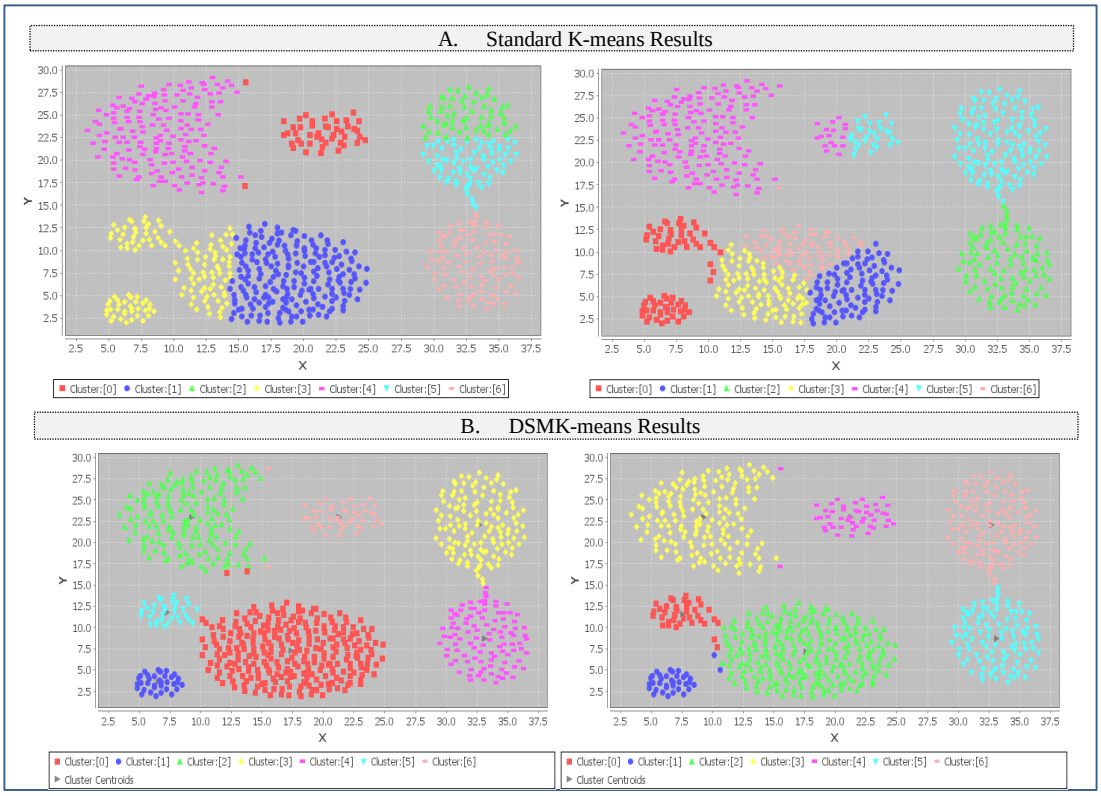

Figure 4.13: Results of K-means and DSMK-means with k=7, on Aggregation dataset.

Finally, DSMK-means algorithm is in general an improved clustering algorithm based on standard K-means. It consists of two main stages: split and merge stage, and anti-noise stage; these stages enable the algorithm to detect different clusters with different shapes, sizes and densities. Moreover, DSMK-means is robust to noises. Experiments demonstrate that DSMK-means clustering algorithm outperforms the traditional K-means and BNAKDAM clustering algorithms.

Chapter 5

5. Conclusion

5.1 Conclusion

5.2 Future Work

Chapter 5

5. Conclusion

This Chapter summarizes the thesis, discusses its findings and contributions, points to limitations of the current work, and also outlines directions for future research.

6.1 Conclusion

In this thesis, researcher have introduced two new clustering algorithm: DIMK-means "Distance-based Initialization Method for K-means clustering algorithm", and DSMK-means "Density-based Split-and-Merge K-means clustering Algorithm".

DIMK-means algorithm presents a new way to select initial centroids in K-means algorithm. This initialization method is as fast and as simple as the K-means algorithm itself, which makes it attractive in practice. The main reason of this enhancement is to make K-means less sensitive to the initialization process and to get consistent results every time algorithm runs. Experimental results demonstrate that the modification appears to give efficient performance when dealing with several virtual and real-world datasets, and it is observed that the proposed method has substantially outperformed the standard K-means in terms of both speed and accuracy.

DSMK-means algorithm developed from k-means which suffers from unsatisfactory accuracy when the dataset contains clusters with different complex shapes, sizes, noise and/or outliers. DSMK-means included Split and Merge technique, which are proposed to overcome standard K-means merging, or splitting true clusters when working with datasets contains clusters with different complex shapes. In addition, DSMK-means included Anti-noise technique, which was proposed to overcome the sensitivity of standard K-means algorithm to noise. DSMK-means algorithm includes solutions for cluster with complex shapes and datasets with noisy objects. Experimental results demonstrate that the algorithm gives efficient performance when dealing with several virtual and real-world datasets, and it is observed that the proposed method is able to define clusters with different shapes that K-means cannot define such clusters.

6.2 Future Work

The results of this thesis point to several interesting directions for future work, which should be addressed and further developed to acquire better results with less cost, these points include the following:

- Develop new techniques to identify most suitable initial centroids
- Improve DIMK-means algorithm through getting rid of the assumption of equal number of objects in each cluster of any given dataset.
- The future work can be focused on reducing the time and computation complexity of DSMK-means algorithm.
- Merging DIMK-means and DSMK-means algorithms into one comprehensive algorithm with reduced
- Develop a new technique to identify the number of clusters (K) automatically is one of the interesting challenges as parameter (K) has to be chosen subjectively in standard K-means algorithm instead of having it as an input parameter.

References

[1] wikipedia, "A Review of Image Data Clustering Techniques," April 2012. [Online]. Available: http://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak.

[2] T. Abraham and J. F. Roddick, "Survey of Spatio-Temporal Databases," *GeoInformatica,* vol. 3, March 1999.

[3] D. Birant and A. Kut, "ST-DBSCAN: an algorithm for clustering spatial–temporal data," *Data & Knowledge Engineering,* vol. 60, pp. 208-221, 2007.

[4] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems), Morgan Kaufmann, 3 edition (July 6, 2011), p. 335–391.

[5] M. S. (. V. K. (. Pang-Ning Tan (Author), Introduction to Data Mining, Addison Wesley; 1 edition (May 12, 2005).

[6] A. JAIN, M. MURTY and P. FLYNN, "Data Clustering: A Review," *ACM Computing Surveys,* vol. 1, September 1999.

[7] A. A. e. al, "Density Based k-Nearest Neighbors Clustering Algorithm for Trajectory Data," *Int. J. on Advanced Science and Technology,* vol. 31, pp. 47-57, June 2011..

[8] J. Tou and R. Gonzalez, "Pattern Recognition Principles," *Addison-Wesley, Reading, MA,* 1974.

[9] S. Selim and M. Ismail, "K-means type a lgorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Trans. Pattern Anal,* vol. 6, p. 81–87, Mach 1984.

[10] H. Spath, "Cluster Analysis Algorithms," *Ellis H orwood, Chichester, UK,* 1989.

[11] S. Kaski. [Online]. Available: http://users.ics.aalto.fi/sami/thesis/node9.html.

[12] A. K. Jain, "Data Clustering: 50 Years Beyond K-Means," *Pattern Recognition Letters,* 2009.

[13] D. Barbara, "An introduction to cluster analysis for data mining," *citeulike:4892027,* 2000.

[14] O. M. (Editor) and L. R. (Editor), "Data Mining and Knowledge Discovery Handbook," Springer; 1 edition, September 1, 2005.

[15] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* p. 281–297, 1967.

[16] H. Vinod, "Integer programming and the theory of grouping," *Journal of the American Statistical Association,* vol. 64, p. 506–519, 1969.

[17] T. Zhang, R. Ramakrishnan and M. Linvy, "BIRCH: an efficient data clustering method for very large databases," *Proceeding ACM SIGMOD International Conference on Management of Data,* p. 103–114, 1996.

[18] S. Guha, R. Rastogi and K. Shim, "CURE: an efficient clustering algorithms for large databases," *Proceeding ACM SIGMOD International Conference on Management of Data, Seattle, WA,* p. 73–84, 1998.

[19] C. S. Warnekar and G. Krishna, "A Heuristic Clustering Algorithm Using Union of Overlapping Pattern-Cells," *Pattern Recognition,* vol. 11, no. 2, pp. 85-93, 1979.

[20] E. Schikuta, "Grid-Clustering: An Efficient Hierarchical Clustering Method for Very Large Data Sets," *Proceedings of the 13th International Conference on Pattern Recognition,* vol. 2, pp. 101-105, 1996.

[21] R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," *Proceedings of 1998 ACM-SIGMOD,* pp. 94-105, 1998.

[22] W. Wang, J. Yang and R. Muntz, "STING: a statistical information grid approach to spatial data mining," *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB),* p. 186–195, 1997.

[23] G. Sheikholeslami, S. Chatterjee and A. Zhang, "WaveCluster: a multi-resolution clustering approach for very large spatial databases," *Proceedings of International Conference on Very Large Databases (VLDB'98), New York, USA,* p. 428–439, 1998.

[24] H.-P. Kriegel, P. Kröger, J. Sander and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 1, pp. 231-240, 2011.

[25] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Portland: AAAI Press,* p. 226–231, 1996.

[26] M. Ankerst, M. M. Breunig, H.-P. Kriegel and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure," *ACM SIGMOD,* vol. 28, no. 2, June 1999.

[27] D. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning,* p. 139–172, 1987.

[28] S. Kalyani and K. Swarup, "Particle swarm optimization based K-means clustering approach for security assessment in power systems," *Expert Systems with Applications,* vol. 30, p. 10839–10846, 2011.

[29] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," *ACM-SIAM Symposium on Discrete Algorithms (SODA 2007) Astor Crowne Plaza, New Orleans, Louisiana*, p. 1–11, 2007.

[30] R. Maitra, "Initializing partition-optimization algorithms," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, p. 144–157, 2009.

[31] C. S. Li, "Cluster Center Initialization Method for K -means Algorithm Over Data Sets with Two Clusters," *2011 International Conference on Advances in Engineering*, vol. 24, p. 324 – 328, 2011.

[32] K. Arai and A. R. Barakbah, "Hierarchical K-means: an algorithm for centroids initialization for K-means," *Rep. Fac. Sci. Engrg, Saga Univ.* , vol. 36, 2007.

[33] M. Naldi, R. Campello, E. Hruschka and A. Carvalho, "Efficiency issues of evolutionaryk-means," *Applied Soft Computing*, vol. 11 , p. 1938–1952, (2011) .

[34] T. Su and J. Dy, "A Deterministic Method for Initializing K-means Clustering," *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference*, pp. 784 - 786 , Nov 2004.

[35] R. C. d. Amorim and B. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering," *Pattern Recognition*, vol. 45, p. 1061–1075, 2012.

[36] Z. Huang, D. Zhang and J. Duan, "BNAK-Divide-and-Merge Clustering Algorithm," in *ICISE '09 Proceedings of the 2009 First IEEE International Conference on Information Science and Engineering* , 2009 .

[37] J. C. Barca and G. Rumantir, "A Modified K-means Algorithm for Noise Reduction in Optical Motion Capture Data," in *Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on*, 11-13 July 2007.

[38] M. Muhr and M. Granitzer, " Automatic Cluster Number Selection Using a Split and Merge K-Means Approach," *IEEE Conference Publications, 20th International Workshop on Database and Expert Systems Application*, pp. 363 - 367, 2009.

[39] E. H., "Ruspini," *Numerical methods for fuzzy clustering. InformationScience*, vol. 2, pp. 319-350, 1970.

[40] Y. P. Osmar Zaïane, "Data Clustering Analysis," Department of Computing Science, University of Alberta, Edmonton, Canada, [Online]. Available: http://webdocs.cs.ualberta.ca/~yaling/Cluster/Php/index.php.

[41] S. a. I. P. Unit, "Clustering datasets," School of Computing, University of Eastern Finland, Finland, [Online]. Available: http://cs.joensuu.fi/sipu/datasets/.

[42] U. o. M. A. F. s. f. t. N. S. Foundation. [Online]. Available: http://archive.ics.uci.edu/ml/.

[43] Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/Iris_flower_data_set#cite_note-fisher36-1.

[44] C. analysis. [Online]. Available: http://en.wikipedia.org/wiki/Cluster_analysis#Evaluation_of_clustering_results.

[45] O. Maimon and L. Rokach, Data Mining And Knowledge Discovery Handbook, 1 ed., 978-0387244358, Ed., amazon, 2005 .

[46] H. Bozdogan, "Akaike's Information Criterion and Recent Developments in Information Complexity," *Journal of Mathematical Psychology*, vol. 44, p. 62–91, 2000.

[47] [Online]. Available: http://en.wikipedia.org/wiki/Akaike_information_criterion.

[48] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6(2), pp. 461-464, 1978.

[49] Y. Z. a. G. Karypis, "Criterion functions for document clustering," *Technical report, Department of Computer Science, University of Minnesota / Army HPC Research Center*, 2002.