CLUSTERING USING GENETIC ALGORITHM

Project submitted to

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

In partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATIONS, 2018

BY

Avijit Chatterjee

Examination Roll: MCA186018

Registration No: 133680 of 2015-2016

Class Roll No: 001510503018

Under the guidance of

Dr. Nirmalya Chowdhury

Professor, Department of Computer Science Engineering

Jadavpur University, Kolkata-700032

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

TO WHOM IT MAY CONCERN

I hereby recommend that the project entitled "Clustering Using Genetic Algorithm" prepared under my supervision and guidance at Jadavpur University, Kolkata by AVIJIT CHATTERJEE*(Reg. No. 133680 of 2015 – 16, Class Roll No. 001510503018 of 2015-16), may be accepted in partial fulfillment for the degree of Master of Computer Applications in the Faculty of Engineering and Technology, Jadavpur University, during the academic year 2017 – 2018. I wish him every success in life.*

…………………………………………..
Prof. (Dr.) Ujjal Maulik
Head of the Department
Department of Computer Science and Engineering
Jadavpur University, Kolkata – 700032.

……………………………………..…
Prof. (Dr.) Nirmalya Chowdhury
Project Supervisor,
Department of Computer Science and Engineering
Jadavpur University, Kolkata – 700032.

……………………………………………
Prof. (Dr.) Chiranjib Bhattacharjee
Dean, Faculty council of Engg. & Tech.
Jadavpur University, Kolkata – 700032.

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC PROJECT

I hereby declare that this project contains literature survey and original research work by the undersigned candidate, as part of his MASTER OF COMPUTER APPLICATIONS studies. All information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material results that are not original to this work.

NAME: AVIJIT CHATTERJEE

ROLL NUMBER: 001510503018

PROJECT TITLE: **CLUSTERING USING GENETIC ALGORITHM**

SIGNATURE WITH DATE:

JADAVPUR UNIVERSITY
FACULTY OF ENGINEERING AND TECHNOLOGY

CERTIFICATE OF APPROVAL

The forgoing project is hereby accepted as a credible study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

FINAL EXAMINATION FOR
EVALUATION OF PROJECT: 1.________________________

2.________________________

(Signature of Examiners)

ACKNOWLEDGEMENT

I express my honest and sincere thanks and humble gratitude to my respected teacher and guide *Prof. (Dr.) Nirmalya Chowdhury*, Professor of the Department of Computer Science & Engineering, Jadavpur University, for his exclusive guidance and entire support in completing and producing this project successfully. I am very much indebted to him for the constant encouragement, and continuous inspiration that he has given to me. The above words are only a token of my deep respect towards him for all he has done to take my project to the present shape.

I would like to thank *Mr. Debaditya Barman and Mr. Ritam Sarkar* for valuable support and suggestions to the activities of the project.

Finally, I convey my real sense of gratitude and thankfulness to my family members, specially my elder sister, for being an endless source of optimism and positive thoughts; and last but not the least, my father & mother for their unconditional support, without which I would hardly be capable of producing this huge work.

AVIJIT CHATTERJEE

Examination Roll: MCA186018

Class Roll: 001510503018

Registration No: 133680 of 2015 – 2016

CONTENTS

CHAPTER 1: INTRODUCTION TO CLUSTERING

1.1 What Is Cluster Analysis?

Cluster analysis or simply **clustering** is 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 clusterings on the same data set. 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.

Cluster analysis has been widely used in many applications such as business intelligence, image pattern recognition, Web search, biology, and security. In business intelligence, clustering can be used to organize a large number of customers into groups, where customers within a group share strong similar characteristics. This facilitates the development of business strategies for enhanced customer relationship management. Moreover, consider a consultant company with a large number of projects. To improve project management, clustering can be applied to partition projects into categories based on similarity so that project auditing and diagnosis (to improve project delivery and outcomes) can be conducted effectively.

Clustering is also called **data segmentation** [1] in some applications because clustering partitions large data sets into groups according to their *similarity*. Clustering can also be used for outlier detection [2], where outliers (values that are “far away” from any cluster) may be more interesting than common cases. Applications of outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce. For example, exceptional cases in credit card transactions, such as very expensive and infrequent purchases, may be of interest as possible fraudulent activities.

As a branch of statistics, cluster analysis has been extensively studied, with the main focus on *distance-based cluster analysis* [3]. Cluster analysis tools based on *k*-means, *k*-medoids, and several other methods also have been built into many statistical analysis software packages or systems, such as S-Plus, SPSS, and SAS. In machine learning, recall that classification is known as supervised learning because the class label information is given, that is, the learning algorithm is supervised in that it is told the class membership of each training tuple. Clustering is known as **unsupervised learning** because the class label information is not present. For this reason, clustering is a form of **learning by observation**, rather than *learning by examples*. In data mining, efforts have focused on finding methods for efficient and effective cluster analysis in *large databases*. Active themes of research focus on the *scalability* of clustering methods, the effectiveness of methods for clustering *complex shapes* (e.g., nonconvex) and *types of data* (e.g., text, graphs, and images), *high-dimensional* clustering techniques (e.g., clustering objects with thousands of features), and methods for clustering *mixed numerical and nominal data* in large databases.

1.2 Requirement for Cluster Analysis

Clustering is a challenging research field. In this section, you will learn about the requirements for clustering as a data mining tool, as well as aspects that can be used for comparing clustering methods.

The following are typical requirements of clustering in data mining.

Scalability: Many clustering algorithms work well on small data sets containing fewer than several hundred data objects; however, a large database may contain millions or even billions of objects, particularly in Web search scenarios. Clustering on only a sample of a given large data set may lead to biased results. Therefore, highly scalable clustering algorithms are needed.

Ability to deal with different types of attributes: Many algorithms are designed to cluster numeric (interval-based) data. However, applications may require clustering other data types, such as binary, nominal (categorical), and ordinal data, or mixtures of these data types. Recently, more and more applications need clustering techniques for complex data types such as graphs, sequences, images, and documents.

Discovery of clusters with arbitrary shape: Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. Consider sensors, for example, which are often deployed for environment surveillance. Cluster analysis on sensor readings can detect interesting phenomena. We may want to use clustering to find the frontier of a running forest fire, which is often not spherical. It is important to develop algorithms that can detect clusters of arbitrary shape.

Requirements for domain knowledge to determine input parameters: Many clustering algorithms require users to provide domain knowledge in the form of input parameters such as the desired number of clusters. Consequently, the clustering results may be sensitive to such parameters. Parameters are often hard to determine, especially for high-dimensionality data sets and where users have yet to grasp a deep understanding of their data. Requiring the specification of domain knowledge not only burdens users, but also makes the quality of clustering difficult to control.

Ability to deal with noisy data: Most real-world data sets contain outliers and/or missing, unknown, or erroneous data. Sensor readings, for example, are often noisy—some readings may be inaccurate due to the sensing mechanisms, and some readings may be erroneous due to interferences from surrounding transient objects. Clustering algorithms can be sensitive to such noise and may produce poor-quality clusters. Therefore, we need clustering methods that are robust to noise.

Incremental clustering and insensitivity to input order: In many applications, incremental updates (representing newer data) may arrive at any time. Some clustering algorithms cannot incorporate incremental updates into existing clustering structures and, instead, have to recompute a new clustering from scratch. Clustering algorithms may also be sensitive to the input data order. That is, given a set of data objects, clustering algorithms may return dramatically different clusterings depending on the order in which the objects are presented. Incremental clustering algorithms and algorithms that are insensitive to the input order are needed.

Capability of clustering high-dimensionality data: A data set can contain numerous dimensions or attributes. When clustering documents, for example, each keyword can be regarded as a dimension, and there are often thousands of keywords. Most clustering algorithms are good at handling low-dimensional data such as data sets involving only two or three dimensions. Finding clusters of data objects in a highdimensional space is challenging, especially considering that such data can be very sparse and highly skewed.

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 teller machines (ATMs) in a city. To decide upon this, you may cluster households while considering constraints such as the city's rivers and highway networks and the types and number of customers per cluster. A challenging task is to find data groups with good clustering behavior that satisfy specified constraints.

Interpretability and usability: Users want clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied in with specific semantic interpretations and applications. It is important to study how an application goal may influence the selection of clustering features and clustering methods.

10.3 Overview of Basic Clustering Methods

There are many clustering algorithms in the literature. It is difficult to provide a crisp categorization of clustering methods [4] because these categories may overlap so that a method may have features from several categories. Nevertheless, it is useful to present a relatively organized picture of clustering methods. In general, the major fundamental clustering methods can be classified into the following categories, which are discussed in the rest of this chapter.

Partitioning methods: Given a set of n objects, a partitioning method constructs k-partitions of the data, where each partition represents a cluster and $k \leq n$. That is, it divides the data into k groups such that each group must contain at least one object. In other words, partitioning methods conduct one-level partitioning on data sets. The basic partitioning methods typically adopt *exclusive cluster separation*. That is, each object must belong to exactly one group. This requirement may be relaxed, for example, in fuzzy partitioning techniques. References to such techniques are given in the bibliographic notes.

Most partitioning methods are distance-based. Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an **iterative relocation technique**[5] that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are “close” or related to each other, whereas objects in different clusters are “far apart” or very different. There are various kinds of other criteria for judging the quality of partitions. Traditional partitioning methods can be extended for subspace clustering, rather than searching the full data space. This is useful when there are many attributes and the data are sparse.

Achieving global optimality in partitioning-based clustering is often computationally prohibitive, potentially requiring an exhaustive enumeration of all the possible partitions. Instead, most applications adopt popular heuristic methods, such as greedy approaches like the *k-means* and the *k-medoids* algorithms, which progressively improve the clustering quality and approach a local optimum. These heuristic clustering methods work well for finding spherical-shaped clusters in small- to medium-size databases. To find clusters with complex shapes and for very large data sets, partitioning-based methods need to be extended.

Hierarchical methods: A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either *agglomerative* or *divisive*, based on how the hierarchical decomposition is formed. The *agglomerative approach*, also called the *bottom-up* approach, starts with each object forming a separate group. It successively merges the objects or groups close to one another, until all the groups are merged into one (the topmost level of the hierarchy), or a termination condition holds. The *divisive approach*, also called the *top-down* approach, starts with all the objects in the same cluster. In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds. Hierarchical clustering methods can be distance-based or density- and continuity based. Various extensions of hierarchical methods consider clustering in subspaces as well.

Hierarchical methods suffer from the fact that once a step (merge or split) is done, it can never be undone. This rigidity is useful in that it leads to smaller computation costs by not having to worry about a combinatorial number of different choices. Such techniques cannot correct erroneous decisions; however, methods for improving the quality of hierarchical clustering have been proposed.

Density-based methods: Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty in discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of *density*. Their general idea is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold. For example, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise or outliers and discover clusters of arbitrary shape.

Density-based methods can divide a set of objects into multiple exclusive clusters, or a hierarchy of clusters. Typically, density-based methods consider exclusive clusters only, and do not consider fuzzy clusters. Moreover, density-based methods can be extended from full space to subspace clustering.

Grid-based methods: Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. Using grids is often an efficient approach to many spatial data mining problems, including clustering. Therefore, grid-based methods can be integrated with other clustering methods such as density-based methods and hierarchical methods.

In the above four method partitioning method is the most simple method and this thesis is also based on that method. So the following section Partitioning Method is discussed only.

1.4 Partitioning Methods

The simplest and most fundamental version of cluster analysis is partitioning, which organizes the objects of a set into several exclusive groups or clusters. To keep the problem specification concise, we can assume that the number of clusters is given as background knowledge. This parameter is the starting point for partitioning methods. Formally, given a data set, *D*, of *n* objects, and *k*, the number of clusters to form, a **partitioning algorithm**[6] organizes the objects into *k* partitions $k \leq n$, where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters in terms of the data set attributes. Here two types of partitioning technique are discussed.

1.4.1 *k*-Means: A Centroid-Based Technique

Suppose a data set, *D*, contains *n* objects in Euclidean space [7]. Partitioning methods distribute

the objects in *D* into *k* clusters, $C1, \ldots\ldots, Ck$, that is, $Ci \subset D$ and $Ci \cap Cj = \theta$ for

$(1 \leq i, j \leq k)$. An objective function is used to assess the partitioning quality so that objects within a cluster are similar to one another but dissimilar to objects in other clusters. This is, the objective function aims for high intracluster similarity and low intercluster similarity.

A centroid-based partitioning technique uses the *centroid* of a cluster, *Ci* , to represent that cluster. Conceptually, the centroid of a cluster is its center point. The centroid can be defined in various ways such as by the mean or medoid of the objects (or points) assigned to the cluster. The difference between an object $\boldsymbol{p} \in Ci$ and ***ci***, the representative of the cluster, is measured by *dist*(***p***, ***ci***), where *dist*(***x***,***y***) is the Euclidean distance between two points ***x*** and ***y***. The quality of cluster *Ci* can be measured by the **within cluster variation**, which is the sum of *squared error* between all objects in *Ci* and the centroid ***ci***, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in Ci} dist(p, ci)^2$$

where *E* is the sum of the squared error for all objects in the data set; ***p*** is the point in space representing a given object; and ***ci*** is the centroid of cluster *Ci* (both ***p*** and ***ci*** are multidimensional). In other words, for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed. This objective function tries to make the resulting *k* clusters as compact and as separate as possible.

Optimizing the within-cluster variation is computationally challenging. In the worst case, we would have to enumerate a number of possible partitionings that are exponential to the number of clusters, and check the within-cluster variation values. It has been shown that the problemis NP-hard in general Euclidean space even for two clusters (i.e., *k* = 2).Moreover, the problem is NP-hard for a general number of clusters *k* even in the 2-D Euclidean space. If the number of clusters *k* and the dimensionality of the space d are fixed, the problem can be solved in time $O(n^{dk+1}$ log$n$), where n is the number of objects. To overcome the prohibitive computational cost for the exact solution, greedy approaches are often used in practice. A prime example is the k-means algorithm, which is simple and commonly used.

*“**How does the** **k**-**means algorithm work?”*** The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. It proceeds as follows. First, it randomly selects *k* of the objects in *D*, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean. The k-means algorithm then iteratively improves the within-cluster variation. For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration. All the objects are then reassigned using the updated means as the new cluster centers. The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round.

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster’s center is represented by the mean value of the objects in the cluster.

Input:

k: the number of clusters,

D: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

```
(1) arbitrarily choose $k$ objects from $D$ as the initial cluster centers;
(2) repeat
```

(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, that is, calculate the mean value of the objects for each cluster;
(5) **until** no change;

1.4.2 *k*-Medoids: A Representative Object-Based Technique

The *k*-means algorithm is sensitive to outliers because such objects are far away from the majority of the data, and thus, when assigned to a cluster, they can dramatically distort the mean value of the cluster. This inadvertently affects the assignment of other objects to clusters.

CHAPTER 2: INTRODUCTION TO GENETIC ALGORITHM

2.1 What is genetic Algorithm?

Genetic Algorithms (GAs) are adaptive heuristic search algorithm [8] based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomised, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, specially those follow the principles first laid down by Charles Darwin of "survival of the fittest."[9]. Since in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones.

2.2 Why Genetic Algorithm

It is better than conventional AI in that it is more robust. Unlike older AI systems, they do not break easily even if the inputs changed slightly, or in the presence of reasonable noise. Also, in dimensional surface, a genetic algorithm may offer significant benefits over more typical search of optimization techniques. (linear programming, heuristic, depth-first, breath-first, and praxis [10])

2.3 Working Principle of Genetic Algorithms (GAs)

The workability of genetic algorithms (GAs) is based on Darwinian's theory of survival of the fittest. Genetic algorithms (GAs) may contain a chromosome, a gene, set of population, fitness, fitness function, breeding, mutation and selection. Genetic algorithms (GAs) begin with a set of solutions represented by chromosomes, called population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. Further, solutions are selected according to their fitness to form new solutions, that is, offsprings. The above process is repeated until some condition is satisfied. Algorithmically, the basic genetic algorithm (GAs) is outlined as below:

Step I

[Start] Generate random population of chromosomes, that is, suitable solutions for the problem.

Step II

[Fitness] Evaluate the fitness of each chromosome in the population.

Step III

[New population] Create a new population by repeating following steps until the new population is complete.

a) [Selection] Select two parent chromosomes from a population according to their fitness. Better the fitness, the bigger chance to be selected to be the parent.

b) [Crossover] With a crossover probability, cross over the parents to form new offspring, that is, children. If no crossover was performed, offspring is the exact copy of parents.

c) [Mutation] With a mutation probability, mutate new offspring at each locus.

d) [Accepting] Place new offspring in the new population.

Step IV

[Replace] Use new generated population for a further run of the algorithm.

Step V

[Test] If the end condition is satisfied, stop, and return the best solution in current population.

Step VI

[Loop] Go to step 2.

The genetic algorithms performance is largely influenced by crossover and mutation operators.

2.4 Encoding Techniques in Genetic Algorithms(GAs)

Encoding techniques in genetic algorithms (GAs) are problem specific, which transforms the problem solution into chromosomes. Various encoding techniques used in genetic algorithms (GAs) are binary encoding, permutation encoding, value encoding and tree encoding.

2.4.1 Binary encoding

It is the most common form of encoding in which the data value is converted into binary strings. Binary encoding gives many possible chromosomes with a small number of alleles[11].

2.4.2 Permutation encoding

Permutation encoding is best suited for ordering or queuing problems. Travelling salesman[12] is a challenging problem in optimization, where permutation encoding is used. In permutation encoding, every chromosome is a string of numbers in a sequence.

2.4.3 Value encoding

Value encoding can be form number, real number on characters to some complicated objects. Value encoding is technique in which every chromosome is a string of some values and is used where some more complicated values are required.

2.4.4 Tree Encoding

It is best suited technique for evolving expressions or programs such as genetic programming. In tree encoding, every chromosome is a tree of some objects, functions or commands in programming languages. Locator/identifier separation protocol (LISP) programming language[13] is used for this purpose. Locator/identifier separation protocol (LISP) programs can be represented in tree structure for crossover and mutation.

2.5 Selection Techniques in Genetic Algorithms (GAs)

Selection is an important function in genetic algorithms (GAs), based on an evaluation criterion that returns a measurement of worth for any chromosome in the context of the problem. It is the stage of genetic algorithm in which individual genomes are chosen from the string of chromosomes. The commonly used techniques for selection of chromosomes are Roulette wheel, rank selection and steady state selection.

2.5.1 Roulette wheel selection

In this method the parents are selected according to their fitness. Better chromosomes, are having more chances to be selected as parents. It is the most common method for implementing fitness proportionate selection. Each individual is assigned a slice of circular Roulette wheel, and the size of slice is proportional to the individual fitness of chromosomes, that is, bigger the value, larger the size of slice is. The functioning of Roulette wheel algorithm is described below:

Step 1 [Sum]

Find the sum of all chromosomes fitness in the population.

Step 2 [Select]

Generate random number from the given population interval.

Step 3 [Loop]

Go through the entire population and sum the fitness. When this sum is more than a fitness criteria value, stop and return this chromosome.

2.5.2 Rank selection method

The application of Roulette wheel selection method is not satisfactory in genetic algorithms (GAs), when the fitness value of chromosomes differs very much. It is a slower convergence technique, which ranks the population by certain criteria and then every chromosome receives fitness value determined by this ranking. This method prevents quick convergence and the individuals in a population are ranked according to the fitness and the expected value of each individual depends on its rank rather than its absolute fitness. For example, if the best chromosome fitness is 80 percent, its circumference occupies 80 percent of the roulette wheel and then other chromosomes will have minimum chances to be selected. On the other hand, the rank selection first ranks the population according to their fitness and then every chromosome receives ranking. The worst will have fitness 1, the second worst will have a fitness of 2, and the best one will have a fitness value n, where n is the number of chromosomes in the population.

2.5.3 Steady-state selection

This method replaces few individuals in each generation, and is not a particular method for selecting the parents. Only a small number of newly created offsprings are put in place of least fit individual. The main idea of steady-state selection is that bigger part of chromosome should retain to successive population.

2.6 Genetic Algorithms (GAs) Operators

Genetic algorithms (GAs) can be applied to any process control application for optimization of different parameters. Genetic algorithms (GAs) use various operators viz. the crossover, mutation for the proper selection of optimized value. Selection of proper crossover and mutation technique depends upon the encoding method and as per the requirement of the problem.

2.6.1 Crossover

It is the process in which genes are selected from the parent chromosomes and new offspring is created. Crossover can be performed with binary encoding, permutation encoding, value encoding and tree encoding.

2.6.1.1 Binary encoding crossover

In binary encoding, the chromosomes may crossover at single point, two point, uniformly or arithmetically. In single point crossover, a single crossover point is chosen and the data before this point are exactly copied from first parent and the data after this point are exactly copied from the second parent to create new offsprings. Two parents in this method give two new offsprings.

2.6.1.2 Uniform Crossover

In uniform crossover, data of the first parent chromosome and second parent chromosome are randomly copied,

2.6.1.3 Arithmetic Crossover

In arithmetic crossover, crossover of chromosomes is performed by AND and OR operators to create new
Offsprings.

2.6.1.4 Permutation encoding crossover

In permutation encoding crossover, one crossover point is selected. The permutation is copied from first parent chromosome upto the point of crossover and the other parent chromosome is exactly copied to ensure that no
number is left to be put in the offspring. Further, if the number is not yet in the offspring, it is added to the offspring chromosome. Travelling salesman problems and task ordering problems[14] can be easily solved by
permutation encoding.

2.6.1.5 Value encoding crossover

It can be performed at single point, two point, uniform and arithmetic representation as in binary encoding

Technique.

2.6.1.6 Tree encoding crossover

In this type of crossover, one point of crossover is selected in both parent tree chromosomes, which are divided at a point. The parts of tree below crossover point are exactly exchanged to produce new offsprings. The choice of the type of the crossover is strictly depends upon the problem.

2.6.2 Mutation

Premature convergence is a critical problem in most optimization techniques, consisting of populations, which occurs when highly fit parent chromosomes in the population breed many similar offsprings in early evolution time. Crossover operation of genetic algorithms (GAs) cannot generate quite different offsprings from their parents because the acquired information is used to crossover the chromosomes. An alternate operator, mutation, can search new areas in contrast to the crossover. Crossover is referred as exploitation operator whereas the mutation is exploration one. Like crossover, mutation can also be performed for all types of encoding techniques.

2.6.2.1 Binary encoding mutation

In binary encoding mutation, the bits selected for creating new offsprings are inverted. In binary encoding mutation, if the bit 1 is converted into bit 0, it decreases the numerical value of the chromosome, and is known as down mutation. Similarly, if the bit 0 is converted into bit 1, the numerical value of the chromosome increases and is referred as up mutation.

2.6.2.2 Permutation encoding mutation

In permutation encoding mutation, the order of the two numbers given in a sequence are exchanged.

2.6.2.3 Value encoding mutation

In value encoding mutation, a small numerical value is either added or subtracted from the selected values of chromosomes to create new offsprings.

2.6.2.4 Tree encoding mutation

Tree encoding mutation, mutates the certain selected nodes of the tree to create new offspring.

Chapter 3: Data Clustering using Genetic Algorithm

3.1 Introduction

Clustering is the organization of a collection of unlabeled patterns, i.e. a vector of measurement or a point in a multidimensional space, into clusters based on their similarity. Intuitively, patterns within a cluster are more similar to each other than they are to a pattern belonging to a different cluster. Labels associated with clusters are data driven, i.e. they are obtained solely from the data, rather than learning from a training set as supervised classification does. Clustering has been widely used in plenty of applications including data mining, pattern recognition, computer vision, image processing and information retrieval. Since recently many clustering algorithms have been developed. Classical clustering algorithms could be enumerated as k-means, nearest neighbor clustering, spectral clustering[15], self organizing map, fuzzy c-mean clustering, and the list goes on. Among them, genetic algorithm (GA), which proposed early in 1989, attracts many attentions because it performs a globalized search for solutions whereas most other clustering approaches perform a localized search and thus easily get stuck at local optimalities. In a localized search, the new obtained solutions inherit the ones in the previous iteration. Such examples are k-means, fuzzy clustering algorithms, ANNs, annealing schemes, and tabu search. Nevertheless, in GAs, the crossover and mutation operators can produce new solutions that are extremely different from from the previous iteration, that is where the global optimality comes from. Besides, GAs are also inherently parallel, making it possible to implement on parallel hardware so to speed up the computation. Actually, GA is an evolutionary approaches, which applies evolutionary operators and a population of solutions to achieve a global optimal partition. Genetic algorithms include selection, recombination and mutation. Candidate solutions to the clustering problem are encoded as chromosomes, and then a fitness function inversely proportional to the squared error value is applied to determine the chromosomes' surviving likelihood in the next generation. In the clustering problem, the solutions best fitting the data is then chosen according to a suitable criterion. In this project, we use a metric distance function rather than a distance matrix to measure the dissimilarity between the instances. The data attributes are converted to be numerical if it is ordinal, and are normalized using L1 norm in each dimension. Thus each instance is embedded into a K dimensional Euclidean space. The aim of the clustering is to minimize the intra-cluster diversity, we use the mean square error as the evaluation measure. In a genetic algorithm (GA) we use a model of the natural selection in real life, where an initial population of solutions called individuals is randomly generated. The algorithm produces new solutions of the population by genetic operations, such as reproduction, crossover and mutation. The new generation consists of the possible survivors with the highest fitness score, and new individuals estimated from the previous population using the genetic operations. In this report, I am going to give a simple review on genetic algorithms specific for the clustering problem. When designing genetic algorithms for clustering problem, three keys play a significant role: The solution representation, the selection method, the crossover method and the mutation method. The efficiency and effectiveness of the GA depends highly on the coding of the individuals. Naturally, a solution could be either a partitioning table, or a set of cluster centroids. The partitioning table actually is the cluster assignments for each instance in the data set. A cluster centroid has the same dimensions as an instance, and is estimated by averaging all the instances in the entire data set which correspond to the particular cluster. Two parent selecting ways are used in this project: a probability-based roulette wheel method and an elitist way with different crossover rate. In the roulette wheel selection, a parent is chosen to according to a probability estimated from the mean squared error. In the elitist method, a set of best solutions are accepted while the rest are dropped. In the crossover phase, six methods are employed, including random crossover, centroid distance, pairwise crossover, largest partitions, pairwise nearest neighbor (PNN), and iterative shrinking (IS). Among them PNN and IS are hybrid methods, which adopt few steps of the conventional k-means clustering algorithm. To improve the clustering performance for the first four crossover methods, it is meaningful to produce new solutions by crossover and then fine-tuned by a partial k-means algorithm.

3.2 K-means Clustering

The k-means clustering method is broadly used and well considered clustering technique (Mac Queen, 1967). Here dataset D is given which has n patterns in real d-dimensional space, Rd, and an integer k, the problem is to resolve a set of k patterns in Rd, called k centers such that the mean squared distance from each pattern to its closest center is minimized. This measure is often called the squared-error distortion (Jain and Dubes, 1988). A standout amongst the most mainstream heuristics for tackling the k-implies issue depends on a basic iterative system for discovering a locally insignificant key, also called Lloyd's k-means algorithm (S. P. Llyod, 1982). The Lloyd’s k-means clustering method is simply called as k-means clustering method. Lloyd's calculation depends on the basic perception that the best arrangement of an inside is at the mean of the joined group. Every group is represented by its mean. The methods must be discovered in a manner that the rule group is represented by its mean. The means have to be found such that the criterion J = Pk j=1PX∈Cj ||X −Mj ||2 is minimized, where Mj is the mean of the cluster Cj . Note that, finding a globally optimal solution for this problem is known to be a NP-hard problem i.e. difficult to solve in polynomial time. But it is still well suitable for cluster formation. (Even for k = 2).

3.2.1 The Basic K-means Algorithm

The k-means clustering procedure is one of the most straightforward algorithms. There are some information points, A =(X1… Xn) that are used to browse and focus on this information. K is number of clusters to be formed and is user parameter; K also depicts number of centroids to group the data. Every point is then allocated to adjacent centroid. For some, we need to arrange gathering of focused information and distribute every point to one group.

The arrangement is to randomly select cluster centre, one for every cluster. The cluster center is then redesigned taking into account method for every gathering which allocate as another centroid. The task is repeated and updated in every iteration until no point changes; "No point change" implies that new centroids are not formed.

K-means formally described by Algorithm:

Input: k: the digit of clusters,

A: data set of n size.

Output: An arrangement of k clusters.

Routine:

1. Selection of k items from A (initial cluster centroid)

2. Repeat until no change

Each item is allocated to the closest cluster to its nearest. (Distance of each item is calculated from selected cluster centroid using sum of squared error)

Recalculate new cluster centroids

Display the final generated clusters.

There are two essential steps in K-means which are as follows:

- Assigning points to closet centroid

- Centroid and objective function

3.2.1.1 Assigning points to closest centroid:

Proximity measure is used to allocate objects to closest centroid Euclidean distance is a common measure of closeness used in data prints. Sometimes the calculating similarity measure of each point is time consuming; in Euclidean space (Pang et. al., 2006). The sum of the squared error (SSE) is cluster evaluation method. SSE formula is formally defined as follow:

$$\boldsymbol{SSE = \sum_{i=1}^{K} \sum_{x \in ci} dist(Ci, x)^2}$$

Where dist is the distance calculated using standard Euclidean method.

Table describes each symbol of formula:

Symbols	Description
x	An object
C_i	The ith Cluster
c_i	The centriod of Cluster i
c	The centriod of all points
m_i	The number of object in ith Cluster
m	The number of object in dataset
K	The number of Cluster

3.2.1.2 Centroid and objective function

Centroid is a variable influence from the clustering evaluation measure. For example measuring distance, minimize the squared distance of each point to closest centroid. This is the goal of clustering that depends on the proximity of the point to another, which is expressed by an objective function. K means is center based approach and has unsupervised nature. Apart from being simple to understand, there are several more issues with which the algorithm suffers. The upcoming sections discuss the sensitive issues of k-means algorithms that lead to further research in this area.

3.3 Advantages of Genetic Algorithm

GA has gotten impressive consideration because of its potential as a novel advancement strategy. There are three noteworthy favorable circumstances in applying Genetic Algorithms to advancement issues.

- GA does not have much mathematical requirements. Due to their developmental nature, GA will hunt down arrangements without looking to the particular internal working of the issue. It can deal with any sort of target capacities and imperatives (i.e. direct or nonlinear) characterized on discrete, nonstop or blended inquiry spaces Due to their evolutionary nature, GA will search for solutions without looking to the specific inner working of the problem. It can work on discrete, continuous or mixed search spaces. The evolution operators make GA very effective at performing the global search.

- GA provides a great flexibility to hybridize with domain dependent heuristics to make an efficient implementation for a specific problem.

3.4 Genetic Algorithm in K-means Clustering Technique

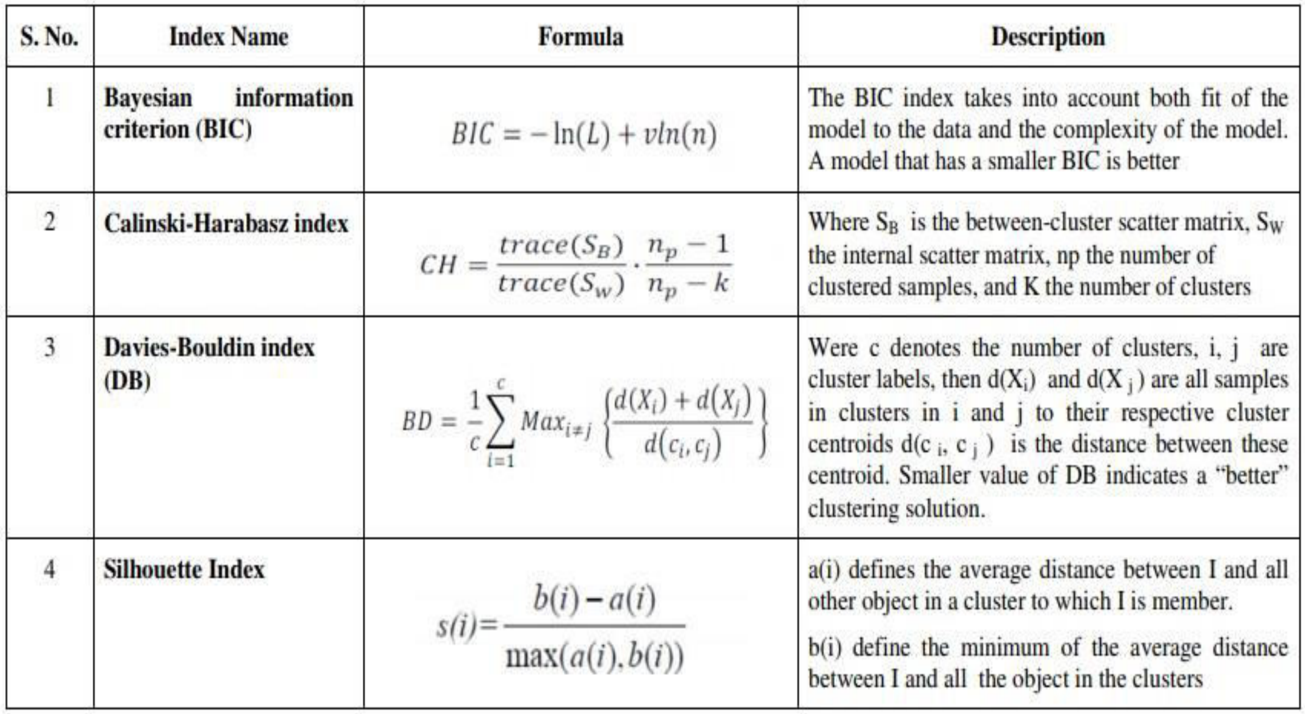
The genetic algorithm is computational strong and scales large variations. This nature into researches to adopt GA in enhancing clustering techniques. Genetic algorithm is one of the best commonly used evolutionary algorithm technique (performs global search) to find the solution to a clustering problem. It has been proved that GA was able to determine the best initialization of clusters as well as was able to optimize initial parameters. GA defines a population of randomly generated individuals. These individuals participate in the generation of new and better offspring using mutation/crossover. Decision regarding better offspring’s / individuals is realized through fitness function. The greatest advantage of genetic algorithms is that the fitness function can be altered to change the behavior of the algorithm. The representation of individual or chromosomes has wide variety. Traditionally, the solutions are represented using fixed length strings, especially binary strings, but alternative encodings have been developed. In clustering algorithm, GA has been applied in two ways: either as a combination or before the clustering algorithm. In both the approach’s k- mean constructed significantly high quality clusters. The literature review revealed that, the major focus of GA based algorithm was to generated high quality clusters in optimized time. Few have also designed the algorithm in multi objective optimization form for their problems which are typical to understand and implement. But unable to give generalized algorithms or deciding the criteria for value of k. While every article that presents a new clustering technique shows its superiority to other techniques, it is hard to judge how well the technique will really do. Thus the two important limitation of k -means are still opened for further exploration for study. In the current research, the focal point was to apply GA as an initial centroid selection tool and to study the performance of improved k -means clustering. In literature, the applications of GA based k means have been tested on standard datasets. But educational dataset specifically out of school children problem has not been investigated. The focal point of the current research was to develop a proper system to study out of school children problem using basic k- means and improved k-means (GA with k-means). Thus the approach in developing a new algorithm was problem specific and criteria of selection or initial centroid was influence from the nature of domain. In short the fitness function in GA has been defined according to the problem domain. Apart from identifying preferable technique for out of school children problem, there is always a need to analyze quality of clusters. There are several techniques to measure the quality of the clusters and performance of clustering. The current research has used these techniques to study the behavior of resultant clusters and clustering's overall performance. The measuring or validation techniques have been listed in next section.

3.5 Validation Techniques for Clustering

Evaluation of clustering results is a challenging task. There are two well known criteria external criteria and internal criteria. In external criteria, the results are judged considering pre-specified structure. In internal criteria, the results are judged considering data intrinsic nature. The use of validity indexes are common approach for evaluation of clustering. And fall in internal criteria selection of indexes depends on information available.

3.5.1 Internal Validation

The first criterion is based on assessing properties of the resulting clusters, like denseness compactness, separation and roundness. This approach is called internal validation and extra information about the data is not mandatory. Table 3.2 describes different internal validation indexes. But in current research, Silhouette index has been used. The literature review states that silhouette index[16] gives slightly more accurate results than other internal index and a promising coefficient to identify value of K.



S. No.	Index Name	Formula	Description
1	**Bayesian information criterion (BIC)**	$BIC = -\ln(L) + vln(n)$	The BIC index takes into account both fit of the model to the data and the complexity of the model. A model that has a smaller BIC is better
2	**Calinski-Harabasz index**	$CH = \frac{trace(S_B)}{trace(S_w)} \cdot \frac{n_p - 1}{n_p - k}$	Where S_B is the between-cluster scatter matrix, S_W the internal scatter matrix, np the number of clustered samples, and K the number of clusters
3	**Davies-Bouldin index (DB)**	$BD = \frac{1}{c}\sum_{i=1}^{c} Max_{i \neq j} \left\{ \frac{d(X_i) + d(X_j)}{d(c_i, c_j)} \right\}$	Were c denotes the number of clusters, i, j are cluster labels, then $d(X_i)$ and $d(X_j)$ are all samples in clusters in i and j to their respective cluster centroids $d(c_i, c_j)$ is the distance between these centroid. Smaller value of DB indicates a “better” clustering solution.
4	**Silhouette Index**	$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$	a(i) defines the average distance between I and all other object in a cluster to which I is member. b(i) define the minimum of the average distance between I and all the object in the clusters

A well-balanced coefficient, the silhouette width which has demonstrated great execution in investigations, was presented by Kaufman and Rousseeuw (2009)[17]. The idea of outline width includes the distinction between the inside of bunch closeness and separation from the rest. The range of silhouette width is from -1 to 1.the value of 0: object can be assign to other cluster if close to -1 : object is misclassified. If close to 1: object is classified well. A clustering can be characterized by the average silhouette width of individual objects. Average silhouette width of clustering is one of the popular criteria for identifying value of k. For example if A is a dataset and A’(k) is average silhouette width of clustering at some k (no of clusters). Than SC is defined as max [A’(k)], from range of k is chosen.(k=1…n).The k will be registered suitable when highest A’(k) is obtained. If the silhouette width ranges from 0.71-1.00 it means that it is an excellent split. If it ranges from 0.51-0.70 it means it is reasonable structure has been found, if it is

0.26- 0.50 it is a weak structure. it could be artificial and if < 0-0.25 it is horrible split (Boros M, 2008).

3.5.2 External Validation

The second criteria, called external or extrinsic validation, compares the partition generated by the clustering algorithm to the true partition of the data. It is direct evaluation measure but is expensive for large dataset. The F-measure, purity, entropy, NMI etc are external validation measure .Literature study does not give any comparative study about the best external measure. Since purity and entropy are simplest extrinsic metrics and are most popular measure for cluster evaluation. Thus chosen in the current research as an evaluation measure there are various type of external validation F-measure, NMI measure[18], Rand statistics[19], Jaccard coefficient[20] etc. Purity is a measure to identify parentage of objects of a single class in a cluster. Purity focuses on the frequency of the most common class into each cluster entropy is reverse of purity. It uses extra knowledge about class labels. The entropy utilizes outside data class names as a part of this case. Purity ranges from 0-1.Larger purity depicts high quality cluster. The entropy is also ranges from 0 to 1. However entropy depicts good clusters loser to 0. Relative measures are also used to study cluster validation. But internal and external measures are widely popular for validation. In current research the silhouette index, purity and entropy measures are used to compare clustering algorithms.

Chapter4: Experimental Results

The approaches which I used

- Minmax normalization for standardization

- Davies–Bouldin index for evaluation of each cluster

IN GENETIC :

* Rank based selection

* One point crossover

INPUT

data which I analysis them is Iris

* **data/iris.csv** have 3 column for 3 dimention

* **config.txt** contain control parameters

The control parameters are as follows:

* kmax : maximum number of clusters

* budget : budget of how many times run GA

* numOfInd : number of Individual

* Ps : probability of ranking Selection

* Pc : probability of crossover

* Pm : probability of mutation

Control Parameters	**Value**
kmax	3
budget	165
numOfInd	20
Ps	0.2
Pc	0.8
Pm	0.2

OUTPUT

- **norm_data.csv** is normalization data

- **cluster_json** is centroid of each cluster

- **result.csv** is data with labeled to each cluster

ANALYSIS

- the accuracy of GA on K-means : 85.33%

GRAPHICAL OUTPUT

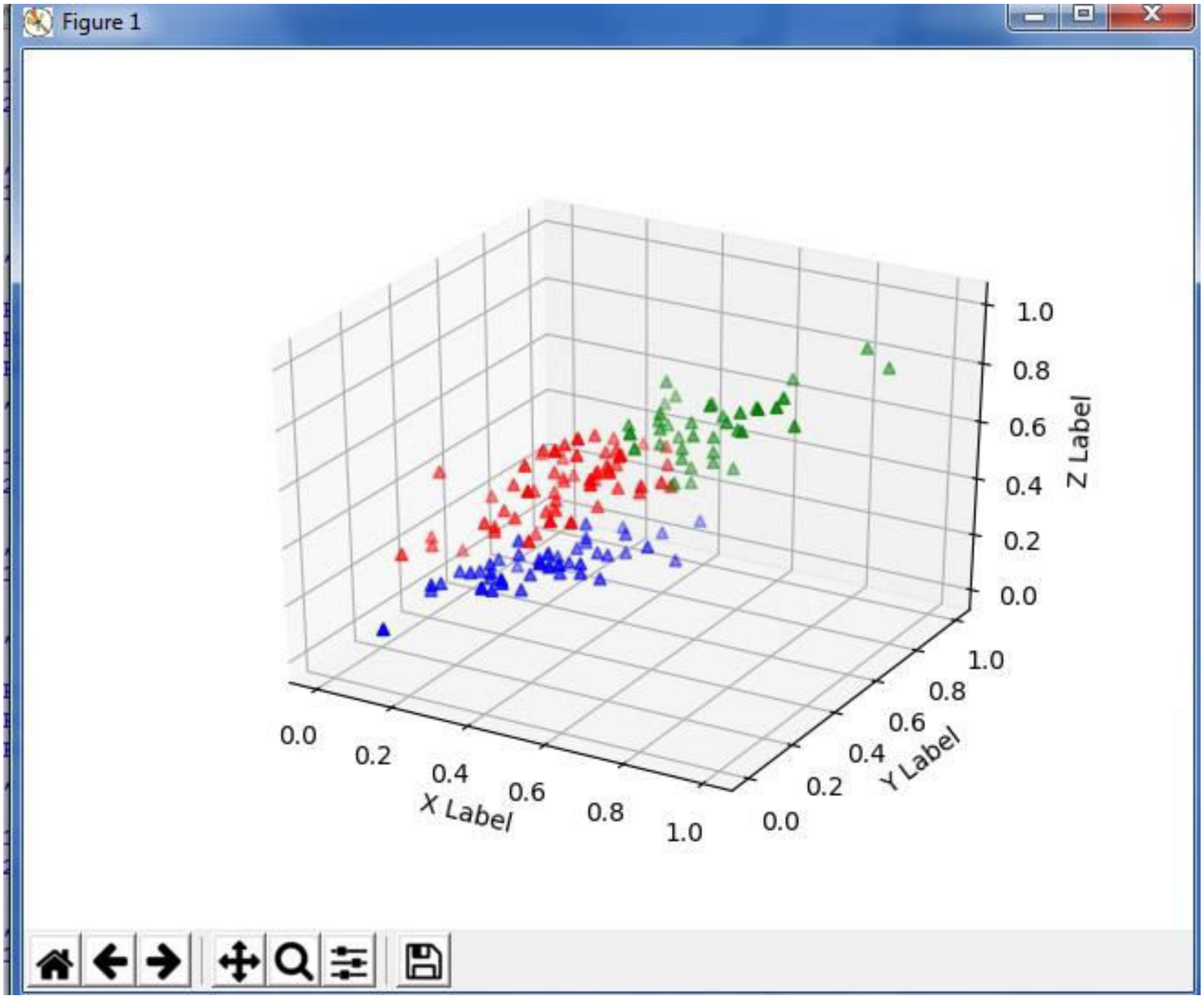


Figure 1
1.0
0.8
0.6
0.4
0.2
0.0
Z Label
0.0
0.2
0.4
0.6
0.8
1.0
X Label
0.0
0.2
0.4
0.6
0.8
1.0
Y Label

CHAPTER 5 : CONCLUTION AND SCOPE OF FURTHER WORK

Conclusion

K-means clustering is center based partitioning clustering .It is simple, fast and unsupervised approach. But it suffers from diverse limitations like initial centroid problem, selection of appropriate value of K and so on. Initial clustering problem is most critical issue out of all limitations. Several improved versions of Kmeans are suggested by different researchers. Few of them have used optimization techniques for handling the start up seed problem. Also different cluster validation measures are incorporated with the experiments to analyze the performance of developed algorithms and resultant cluster quality. Genetic algorithm is one of well known optimization algorithm used to overcome K-means weakness. In clustering algorithm, GA has been applied in two ways: as a combination or before the clustering algorithm. In both the approach's k- mean constructed significantly high quality clusters. The literature review revealed that, the major focus of GA based algorithm was to generate high quality clusters in optimized time. Some has also designed the algorithm in multi objective optimization form for their problems that are multifarious to understand and implement. Therefore, unable to give a generalized clustering algorithm or a method to decide the value of k. Every article

that presents a new clustering technique shows its superiority to other techniques, it is hard to judge how well the technique will really do. Thus the two important limitation of k -means are still opened for further exploration for study. This chapter summarized the K-means clustering, significance of GA based clustering and their scope in educational data sets with effective cluster quality measures. The current research focuses to develop a proper system to study out of school children problem using basic k- means and improved k-means (GA with k-means) algorithms. Also to identify suitable measures to study performance of the system, clustering algorithms and quality of clusters. The next chapter discusses the methodology of the current research.

Scope for Further Work

Recent research has shown that clustering techniques that optimize a single objective may not provide satisfactory result because no single validity measure works well on different kinds of data sets. Then we can further develop a multiobjective fuzzy genetic clustering method that optimizes multiple validity measures simultaneously. User can chose any partitioning result from the resultant set of non dominated solutions according to the problem requirements.

APPENDIX

References

[1] https://en.wikipedia.org/wiki/Data_segment

[2] Outlier Detection and Removal Algorithm in K-Means and Hierarchical Clustering (Anwesha Barai (Deb), Lopamudra Dey*)

[3] https://link.springer.com/article/10.1007/BF00192485

[4] https://www.researchgate.net/publication/233721790_Fuzzy_and_Crisp_Clustering_Methods_Based_on_The_Neighborhood_Concept_A_Comprehensive_Review

[5] http://www.tqmp.org/RegularArticles/vol09-1/p015/p015.pdf

[6] https://www.slideshare.net/Krish_ver2/32-partitioning-methods

[7] https://en.wikipedia.org/wiki/Euclidean_space

[8] https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html

[9] https://en.wikipedia.org/wiki/Survival_of_the_fittest

[10] https://link.springer.com/article/10.3758/BF03203605

[11] https://www.cse.unsw.edu.au/~billw/cs9414/notes/ml/05ga/05ga.html

[12] https://simple.wikipedia.org/wiki/Travelling_salesman_problem

[13] https://en.wikipedia.org/wiki/Lisp_(programming_language)

[14] https://discuss.gradle.org/t/task-ordering-problem-mustrunafter-finalizer-dependencies/14035

[15] https://en.wikipedia.org/wiki/Spectral_clustering

[16] https://en.wikipedia.org/wiki/Silhouette_(clustering)

[17] Kaufman and Rousseeuw (2009)

[18] https://en.wikipedia.org/wiki/National_Measurement_Institute,_Australia

[19] https://en.wikipedia.org/wiki/Rand_index

[20] https://en.wikipedia.org/wiki/Jaccard_index