A Genetic Algorithm Approach for Semi-Supervised Clustering Algorithm

M. Pavithra[1], Dr.R.M.S.Parvathi [2.]
Assistant Professor, Department of C.S.E,
Jansons Institute of Technology, Coimbatore, India[1].
Dean- PG Studies, Sri Ramakrishna Institute of Technology, Coimbatore, India[2].

ABSTRACT

A semi-supervised clustering algorithm is proposed that combines the benefits of supervised and unsupervised learning methods. The approach allows unlabeled data with no known class to be used to improve classification accuracy [2]. The objective function of an unsupervised technique, e.g. K-means clustering, is modified to minimize both the cluster dispersion of the input attributes and a measure of cluster impurity based on the class labels. Minimizing the cluster dispersion of the examples is a form of capacity control to prevent over fitting [4]. For the output labels, impurity measures from decision tree algorithms such as the Gini index can be used. A genetic algorithm optimizes the objective function to produce clusters. Experimental results show that using class information improves the generalization ability compared to unsupervised methods based only on the input attributes [6]. Training using information from unlabeled data can improve classification accuracy on that data as well. Genetic Algorithms (GAs) have been widely used in optimization problems for their high ability in seeking better and acceptable solutions within limited time. Clustering ensemble has emerged as another flavour of optimal solutions for generating more stable and robust partition from existing clusters [1]. GAs has proved a major contribution to find consensus cluster partitions during clustering ensemble. Currently, web video categorization has been an ever challenging research area with the popularity of the social web. In this paper, we propose a framework for web video categorization using their textual features, video relations and web support [3]. There are three contributions in this research work. First, we expand the traditional Vector Space Model (VSM) in a more generic manner as Semantic VSM (S-VSM) by including the semantic similarity between the features terms [5]. This new model has improved the clustering quality in terms of compactness (high intra-cluster similarity) and clearness (low inter-cluster similarity). Second, we optimize the clustering ensemble process with the help of GA using a novel approach of the fitness function. We define a new measure, Pre-Paired Percentage (PPP), to be used as the fitness function during the genetic cycle for optimization of clustering ensemble process [7]. Third, the most important and crucial step of the GA is to define the genetic operators, crossover and mutation. We express these operators by an intelligent mechanism of clustering ensemble. This approach has produced more logical offspring solutions [9]. Above stated all three contributions have shown remarkable results in their corresponding areas. Experiments on real world social-web data have been performed to validate our new incremental novelties [8].

KEYWORDS: Data Mining,Clustering, Clustering Ensemble, Pair wise Constraints, Genetic Algorithm, Semantic Similarity.

I. INTRODUCTION

Clustering can be used for this task. In the medical field, clustering of data can be used to determine if a drug provides greater benefits to a certain group of patients. Grouping of information is used in the engineering field to determine what factors lead to the failure of a component in a system [1]. And in marketing, data clustering can give a clearer picture of how to focus an advertising campaign to the proper audience. This paper will discuss the use of Genetic Algorithms (GAs) for the task of clustering data [3]. In particular, the application of GAs for clustering on very large data sets, such as image data sets, will be addressed. The running time for most clustering GAs becomes quite large as the size of the input data set increases [2]. We propose an efficient genetic algorithm for clustering on very large image data sets.The diversity of applications for clustering has lead to many problem definitions [5]. The objective of all clustering algorithms is to divide a set of data points

into subsets so that the objects within a subset are similar to each other and objects that are in different subsets have diverse qualities [6]. The fact that there are many different methods used to quantify the similarity and diversity of data points leads to the many different variations of the problem [8]. For our research, we defined the clustering problem as the task of dividing an input data set into a desired number of subgroups so that the Euclidean distance between each data point and its corresponding cluster centre is minimized. This is a very common method of defining the clustering problem [10]. The total of the distances of each point to its cluster centre is known as the total distance measurement of the clustering [9].

Clustering analysis works to classify a set of unlabeled instances into groups such that instances in the same group are more similar to each other, while they are more different in different groups [5]. In its traditional literature, clustering analysis was considered as an unsupervised method for data analysis, which performs under the condition that no information is available concerning memberships of instances to predefined groups [1]. However, it was known that some background knowledge such as instance-level constraints can be obtained easily in many real-world applications and several recent studies have also shown that these instance-level constraints can significantly increase accuracies of a variety of clustering algorithms [9]. Clustering analysis under the condition that some limited instance-level constraints are incorporated for guiding the clustering of the data was termed as semi-supervised clustering with instance-level constraints, which has become one of the most active research topics in the areas of pattern recognition, machine learning and data mining [3]. Semi-supervised clustering with instance-level constraints has gained some real-world applications such as GPS-based map refinement, person identification from surveillance camera clips and landscape detection from hyper spectral data [2]. However, semi-supervised clustering with instance-level constraints is not exempt from any drawbacks. One disadvantage of semi-supervised clustering with instance-level constraints is that instance-level constraints tend to split the search space of the optimal clustering solution into pieces that compounds the difficulty of the search task. Whereas commonly-used hill-climbing search methods can only guarantee a local optimal clustering solution [10]. The above disadvantage of semi-supervised clustering with instance-level constraints motivates us to adopt genetic algorithms to perform the search task [11].

Evolutionary Computation (EC) is a field of computer science that uses biological processes as a model for solving computer-based problems [2]. Genetic Algorithms (GAs), first proposed by John Holland in the 1960s, are a category of EC that use concepts derived from evolution. Proper application of a GA finds a balance between exploration and exploitation of a given optimization problem's search space. A good overview of how to design a GA is given in [11]. It shows the structure that is used by GAs. First, a population of chromosomes is created an initialized. These chromosomes each contain a collection of genes and each gene has a value (called an allele) [4]. A single chromosome is an encoded version of a solution to the problem that the GA is attempting to optimize. The GA performs exploration/exploitation of the problems search space by evolving the population of chromosomes through a series of generations [6]. During each generation of the GA, parent chromosomes are selected from the population. These parent chromosomes are combined to form children chromosomes and then the child chromosomes are mutated [9]. In a generational type GA, an entirely new population for each generation is formed by creating multiple child chromosomes. For a steady state GA, the child chromosomes are used to replace members of the current population but a new population is not formed during each generation [10]. A very important step in the GA is the selection of parents for the next generation of chromosomes. In order to provide a guided search, which is appropriate for the given optimization problem, the selection of parents needs to be based on the quality of the solution that their chromosomes represent [12]. A property called fitness is used to quantify the quality of a given solution and a fitness function is used to calculate thefitness value of each chromosome in a given population before parent selection is made [4]. A variety of different selection methods are used by GAs but they all use the principle that higher fit chromosomes are more likely to be chosen as parents. This fitness selection provides the GA direction for the search of an optimization problems search space [2].

II. RELATED WORK

Traditional clustering algorithms are unsupervised under the condition that no information is available concerning memberships of instances to predefined groups [1]. However, if no information about memberships

of instances is available, clustering analysis is an ill-posed combinatory optimization problem and no single clustering algorithm is able to achieve high quality clustering solutions for all kinds of data sets. A large number of studies have been concentrated on improving the robustness and stability of clustering algorithms [3]. Among them is semi-supervised clustering algorithms that incorporate some prior background knowledge about memberships of instances into the original framework of traditional unsupervised clustering algorithms. It should be noted that these prior background knowledge can sometimes be obtained naturally from application domains without accessing any human interaction [5]. For example, to segment movies such that all the frames in which the same actor appears are grouped. Due to the continuous nature of most movies, faces extracted from successive frames in roughly the same location can be assumed to come from the same person. Another example is to segment images using clustering algorithms [4]. Two pixels have a high probability to be grouped together if they are spatially connected. Many recent studies have demonstrated that these prior background knowledge can significantly improve accuracies of clustering algorithms [6]. Clustering algorithms incorporate these prior background knowledge in a constrained format, which may come from several diff erent sources such as partial labels, instances relationships and spatial contiguity [8].

Clustering analysis works to classify a set of unlabeled instances into groups such that instances in the same group are more similar to each other, while they are more diff erent in diff erent groups []. Many feasible approaches have been proposed to classify a set of unlabeled instances into groups and most of them belong to the following three categories: data partitioning, hierarchical clustering and model-based clustering [7]. This paper is mainly interested in K-means clustering algorithm, which is one of the most famous data partitioning algorithms. Therefore, in this paper data clustering algorithm works to search for a partition of all instances such that the minimization of the within-cluster variation can be achieved [3]. A variate of this kind of approaches is semi-supervised clustering with penalty that works to penalize clustering solutions according to the degrees in which they violate the given instance-level constraints [8]. The second one is to learn a distance metric from all available instance-level constraints such that instances in the learned distance space are more suitable for the clustering of the data [9]. For more information about semi-supervised clustering algorithms with instance-level constraints.Genetic algorithm is a class of heuristic search algorithms based on the mechanism of nature selection [2]. It has been widely applied into the area of data analysis such as data clustering, feature selection and machine vision. In this paper, we explore the genetic algorithm to optimize the objective function [1].

The most frequently used criterion is the distance between two clusters that are to be joined. The algorithm needs also a stopping criterion to prevent the agglomeration process from continuing up to the point where all instances belong to one cluster [1]. The concept of agglomerative clustering can be viewed as a greedy search method in which at each step we look for the currently optimal arrangement of clusters given the condition that two clusters must be joined [2]. Let us assume that we look for the two clusters with a minimal average distance between members of both clusters. To find the currently optimal join the algorithm examines all possible pairs of clusters [4]. We could easily think of an extension of agglomerative clustering in which the algorithm examines a chosen pair of clusters and makes a decision if they should or should not be joined. The extension limits the computational and operational cost of the processing step; however, the local character of the decision can easily lead the algorithm to local optimum [3]. The objective of our work is to implement the ideas of such extended, local decision based, agglomerative clustering in the form of a genetic algorithm [7].The above examples indicate that a GA may be used to optimize one or more clustering quality scores to produce high quality clusters. Each of the presented approaches assumes that the complete dataset can be represented as a single individual and processed at once [9]. The main objective of our work was to reduce the size of an individual and the computation cost of a single fitness score to make the algorithm feasible given the real sizes [8].

III. GENETIC ALGORITHMS FOR CLUSTERING DATA

Using a GA to solve data clustering problems is not a new idea. GAs has been successfully implemented for various clustering problems using different chromosome encoding schemes and fitness functions. In [2] a GA is used to solve the clustering problem for a data set of geographical data. Each data point in the input data set is assigned a unique integer value from 1 to n, where n is the total number of data points in the input set [5]. The

chromosomes in a population contain one gene for each data point that is to be clustered and the allele values of the genes designate the assignment of all n points to the desired number of clusters. The total length of a chromosome is n [7]. The fitness function used in the GA mimics the objective function of the k-means algorithm, which is shown in. The algorithm described in [3] uses a multistep procedure. The authors refer to this procedure as a semi supervised form of learning. A GA performs clustering on an input set of data objects so that supervised learning can be applied to predict class labels in the second step [10]. The input for the GA is a set of data objects that have both numeric and label attributes and a desired number of clusters. The goal of the GA is to produce clusters of data objects that minimize cluster dispersion and are as pure as possible in relation to the label attributes. The GA uses a two component fitness function where the first component measures the within [8].

IV. SEMI-SUPERVISED CLUSTERING FOR GENETIC APPROACH

As a base for our semi-supervised algorithm, we use an unsupervised clustering method combined with a genetic algorithm incorporating a measure of classification accuracy used in decision tree algorithms, the GINI index [8]. Here, we examine the clustering algorithm that minimizes some objective function applied to k-cluster centres. In our case, we consider the cluster dispersion and cluster purity. Before the clustering task, each term is assigned with a specific weight that is normalized across all terms [4]. The main objective is to choose the best weights for all terms considered that minimize some measure of cluster dispersion and cluster quality. In out GA algorithm, the fitness function will be the reciprocal of objective function [5]. Typically cluster dispersion metric is used, such as the Davies-Bouldin Index (DBI). DBI uses both the withincluster and between-clusters distances to measure the cluster quality [6]. Let dcentroid (Qk), defined in [8], denotes the centroid distances within cluster. Clustering using K cluster centres partitions the input space into K regions [1]. Therefore clustering can be considered as a K-nary partition at a particular node in a decision tree, and GI can be applied to determine the purity of such partition (cluster purity). In this case, GI of a certain cluster, k, is computed as defined in [12], where n is the number of class, Pkc is the number of points belong to c-th class in cluster k and Nk is the total number of points in cluster k [2].

$$GiniC_k = \quad 1.0 - \sum_{c=1}^{n} \left(\frac{P_{kc}}{N_k} \right)^2$$

$$impurity = \quad \frac{\sum_{k=1}^{K} TC_k \cdot GiniCk}{N}$$

$$\text{f(N,K)} = \quad \text{Cluster Dispersion + Cluster Purity}$$

$$\text{f(N,K)} = \quad DBI + \frac{\sum_{k=1}^{K} TC_k \cdot GiniCk}{N}$$

V. PROPOSED WORK

V a. VECTOR SPACE MODEL REPRESENTATION (VSM)

In this work, we use the vector space model, in which a document is represented as a vector in an n-dimensional space (where n is the number of different words in the collection of documents) [7]. Here, documents are categorized by the words they contain and their frequency []. Before obtaining the weights for all the terms extracted from these documents, stemming and stop word removal is performed. Stop word removal eliminates unwanted terms (e.g., those from the closed vocabulary) and thus reduces the number of dimensions in the term-space [5]. Once these two steps are completed, the frequency of each term across the corpus is counted and weighted using term frequency – inverse document frequency (tf-idf), as described. Weights are assigned to give an indication of the importance of a word in characterizing a document as distinct from the rest of the corpus

[4]. In summary, each document is viewed as a vector whose dimensions correspond to words or terms extracted from the document. The component magnitudes of the vector are the tf-idf weights of the terms. In this model, tf-idf, as described in, is the product [10].

$$\text{tf-idf} = \text{tf}(t,d) \cdot \text{idf}(t)$$

$$\text{idf}(t) = \log_{10}\left(\frac{|D|}{\text{df}(t)}\right)$$

$$\text{sim}(d_i, d_j) = \frac{(d_i d_j)}{(\|d_i\| \cdot \|dj\|)}$$

$$\text{Precision (C,L)} = \frac{|C \cap L|}{|C|}, C \in C_{ALL}, L \in L_{ALL}$$

V b. AGGLOMERATIVE CLUSTERING GENETICALGORITHM (ACGA)

In standard approaches to applying GAs to clustering problems each individual denotes a complete solution. The solutions compete and exchange genetic material evolving toward a well-fitted individual that represents the final partition [3]. In our Agglomerative Clustering Genetic Algorithm (ACGA) each individual represents one cluster. The crossover operator allows two individuals to exchange genetic material of two clusters to locally improve the value of fitness function [4]. The number of individuals is equal to the current number of clusters and the complete solution is represented by the whole population [5]. Given the assumption that a cluster size is small and limited (fine-grained communities) this modification allows the algorithm to calculate the value of the fitness function engaging only a small part of the graph at a time, hence making the processing feasible independently of the size [7]. This section describes modifications of genetic operators required by ACGA. We concentrate on hierarchical agglomerative clustering [11]. Unlike partitioned clustering algorithms that build a hierarchical solution from top to bottom, repeatedly splitting existing clusters, agglomerative algorithms build the solution by initially assigning each document to its own cluster and then repeatedly selecting and merging pairs of clusters, to obtain a single all-inclusive cluster, generating the cluster tree from leaves to root [12].

$$\text{Purity} = \sum_{C \in C_{ALL}} \frac{|C|}{|D|} \cdot P(C,L)$$

$$\text{Precision (EBM)} = \frac{|C(E) \cap C(B)|}{|C(E)|}$$

$$\text{Precision (BEM)} = \frac{|C(B) \cap C(E)|}{|C(B)|}$$

V c. FAST GENETIC K-MEANS ALGORITHM (FGKA)

FGKA maintains a population (set) of Z coded solutions (partitions), where Z is a parameter specified by the user. Each solution is coded by a string a1…aN of length N [2]. Given a solution Sz = a1…aN, we define the legality ratio of Sz, e(Sz), as the number of non-empty clusters in Sz divided by K. Sz is legal if e(Sz)=1, and illegal otherwise [5]. FGKA starts with the initialization phase, which generates the initial population P0 [3]. The population in the next generation Pi+1 is obtained by applying the following genetic operators sequentially: the selection, the mutation and the K-means operator on the current population Pi [6]. The evolution takes place until the termination condition is reached. FGKA, but are considered as the most undesirable solutions by

defining their TWCVs as $+\infty$ and assigning them with lower fitness values [8]. Our flexibility of allowing illegal strings in the evolution process avoids the overhead of illegal string elimination, and thus improves the time performance of the algorithm. In the following, we give a brief description of the three genetic operators [9].

$$p_z = \frac{F(S_z)}{\sum_{z=1}^{Z} F(S_z)} (z = 1, \cdots, Z),$$

$$F(S_z) = \begin{cases} 1.5 * TWCV_{\max} - TWCV(S_z), if\ S_z\ is\ leagal \\ e(S_z) * F_{\min}, otherwise \end{cases}$$

V d. GENETIC K-MEANS ALGORITHM (GKA)

The name genetic K-means algorithm (GKA). We define K-means operator, one-step of K-means algorithm, and use it in GKA as a search operator instead of crossover [2]. We also define a biased mutation operator specific to clustering called distance-based-mutation. Using finite Markov chain theory, we prove that the GKA converges to the global optimum [5]. It is observed in the simulations that GKA converges to the best known optimum corresponding to the given data in concurrence with the convergence result [9]. It is also observed that GKA searches faster than some of the other evolutionary algorithms used for clustering.GKA maintains a population of coded solutions. Thepopulation is initialized randomly and is evolved over generations;the population in the next generation is obtained by applying geneticoperators on the current population [10]. The evolution takes place until aterminating condition is reached. The genetic operators that are usedin GKA are the selection, the distance based mutation and the K-means operator [12]. In this section we explain GKA by specifying thecoding and initialization schemes and, the genetic operators [13].

$$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}$$

```
Mutation(sW)
{ for i = 1 to n
  { if (drand() < Pm)
    { Calculate cluster centers, cj's,
    corresponding to sW;
     for j = 1 to K, dj = d(xi, cj);
     if (d_sW(i) > 0)
       { dmax = max{d1, d2, ..., dK}
         for j = 1 to K,
             pj = (cm dmax - dj)/ sum_{k=1}^{K} (cm dmax - dk)
         sW(i) = a number, randomly selected from
                 {1, 2, ..., K} according to the
                 distribution {p1, p2, ..., pK};
       }
    }
  }
}
```

The algorithm with the above selectionand mutation operators may take more time to converge, since theinitial assignments are arbitrary and the subsequent changes of theassignments are probabilistic [11]. Moreover, the mutation probability isforced to assume a low value because high values of lead tooscillating behaviour of the algorithm [10].

V e. HYBRID GENETIC BASED CLUSTERING ALGORITHM (HGA)

This algorithm, with the cooperation of tabu list and aspiration criteria, has achieved harmony between population diversity and convergence speed [2]. A genetic algorithm was proposed for designing the dissimilarity measure, termed Genetic Distance Measure (GDM) such that the performance of the K-modes algorithm is improved [3]. The combines' genetic algorithm (GA) with simulated annealing to find optimal solution [6]. This algorithm maximized the clustering success by achieving internal cluster cohesion and external cluster isolation [5]. The performance of HGACLUS was compared with other existing clustering methods and was found to be more accurate and robust than other methods [7].

V f. GENETIC WEIGHTED K-MEANS ALGORITHM (GWKMA)

In a general sense, a *k*-partitioning algorithm takes as input a set D = {x_1, x_2⋯, x_n} of *n* objects and an integer *K*, and outputs a partition of *D* into exactly *K* disjoint subsets D_1,⋯, D_K[3]. Denote such a partition by Δ. Each of the subsets is a cluster, with objects in the same cluster being somehow more similar to each other than they are to all subjects in other different clusters [4]. One way to make the determination of Δ into a well-defined problem is to define a cost function which measures the clustering quality of any partitions of a dataset [11]. Each attribute of an object (gene) is expressed as a real number and thus each object may be described by a real number row vector of dimension *d*, where *d* is the number of attributes of an object [12]. Assume that all objects in the dataset have the same number of attributes, i.e. no missing data. Let (x_i,*i* = 1,⋯, *n*) be a dataset of *n* objects. Let x_{ij}denote the *j*th attribute of object x_i. X= (x_{ij}) is called an attribute matrix of object set D [13]. For the predefined number *K* of clusters, the costs function for a weighted k-means clustering technique [6].

$$J_G(\Delta) = \sum_{k=1}^{K} \sum_{x_i \in D_k} (x_i - \overline{m}_k) G (x_i - \overline{m}_k)'$$

where

$$\overline{m}_k = \frac{1}{n_k} \sum_{x_i \in D_k} x_i$$

VI. EXPERIMENTS

In this section, we empirically demonstrate that our proposed semi-supervised clustering for genetic algorithm is both efficient and eff ective.

VII a. DATASETS

The data sets used in our experiments include six UCI data sets1. Here is some basic information of those data sets. Table 5 summarizes the basic information of those data sets.

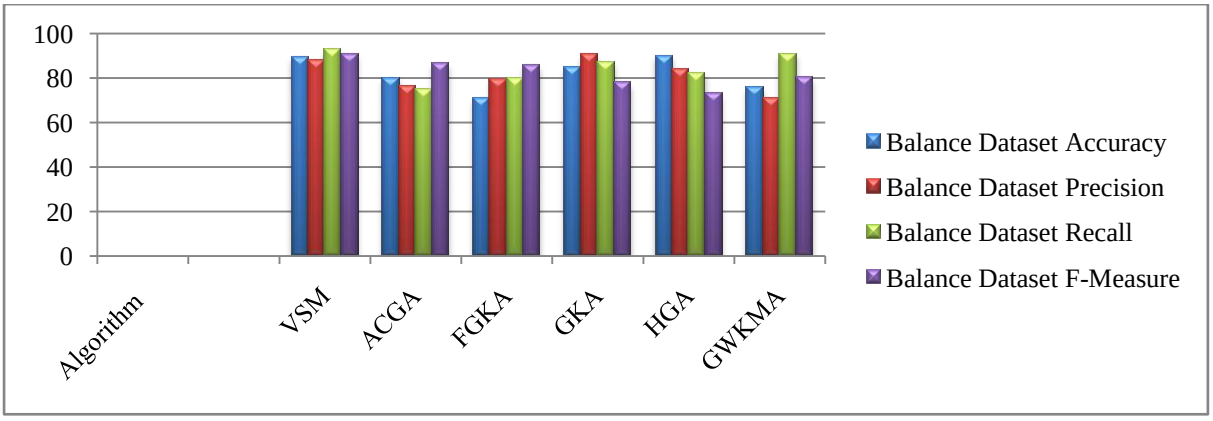
- Balance. This data set was generated to model psychological experimental results. There are totally 625 examples that can be classified as having the balance scale tip to the right, tip to the left, or be balanced.
- Iris. This data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant.
- Ionosphere. It is a collection of the radar signals belonging to two classes. The data set contains 351 objects in total, which are all 34-dimensional.
- Soybean. It is collected from the Michalski's famous soybean disease databases, which contains 562 instances from 19 classes.

Datasets	Size	Classes	Dimensions
Balance	625	3	4
Iris	150	3	4
Ionosphere	351	2	34
Soybean	562	19	35

VIII. EXPERIMENTAL RESULTS

VIII a. BALANCE DATASET RESULTS

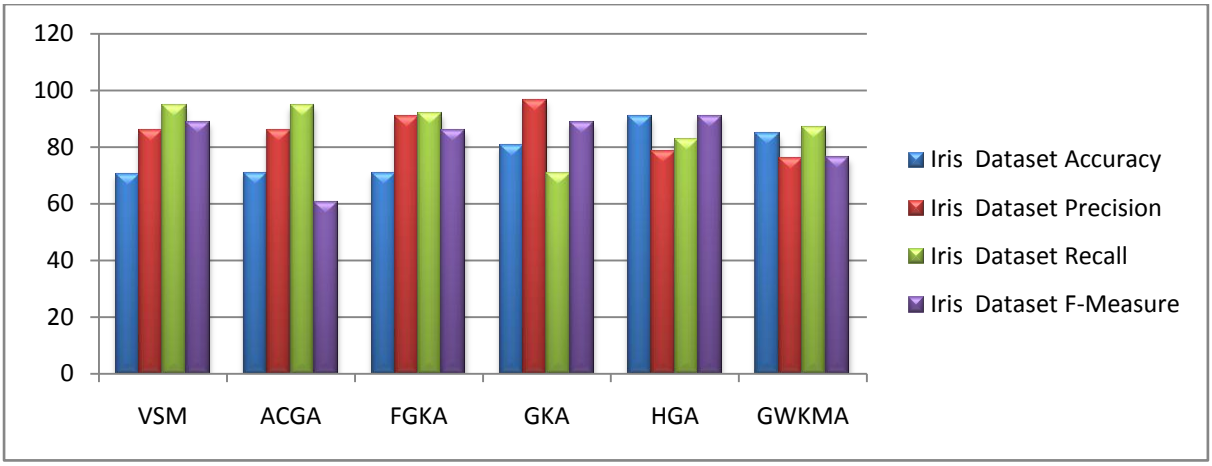
Balance Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
VSM	89.45	87.91	92.77	90.89
ACGA	79.91	76.08	74.78	86.56
FGKA	70.92	79.67	79.89	85.78
GKA	84.67	90.67	86.78	77.67
HGA	90.07	83.66	82.33	72.88
GWKMA	75.66	70.89	90.75	80.34



The above graph shows that performance of Balance dataset. The Accuracy of HGA algorithm is 90.07 which is higher when compare to other five (VSM, ACGA, FGKA, GKA, GWKMA) algorithms. The Precision of GKA algorithm is 90.67 which is higher when compare to other five (VSM, ACGA, FGKA, HGA, GWKMA) algorithms. The Recall of VSM algorithm is 92.77 which is higher when compare to other five (GKA, ACGA, FGKA, HGA, GWKMA) algorithms. The F-Measure of VSM algorithm is 90.89 which is higher when compare to other five (GKA, ACGA, FGKA, HGA, GWKMA) algorithms.

VIII b. IRIS DATASET RESULTS

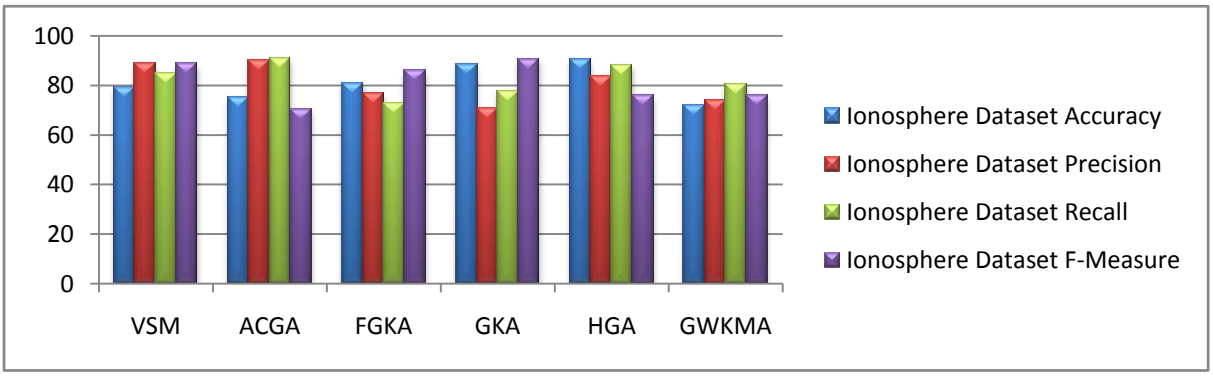
Iris Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
VSM	70.45	85.91	94.77	88.89
ACGA	70.91	86.08	94.78	60.56
FGKA	70.92	90.67	91.89	85.78
GKA	80.67	96.67	70.78	88.67
HGA	90.78	78.76	82.54	90.89
GWKMA	84.56	75.9	87.23	76.12



The above graph shows that performance of Iris dataset. The Accuracy of HGA algorithm is 90.78 which is higher when compare to other five (VSM, ACGA, FGKA, GKA, GWKMA) algorithms. The Precision of GKA algorithm is 96.67 which is higher when compare to other five (VSM, ACGA, FGKA, HGA, GWKMA) algorithms. The Recall of ACGA algorithm is 94.78 which is higher when compare to other five (VSM, HGA, FGKA, GKA, GWKMA) algorithms. The F-Measure of HGA algorithm is 90.89 which is higher when compare to other five (VSM, ACGA, FGKA, GKA, GWKMA) algorithms.

VIII c. IONOSPHERE DATASET RESULTS

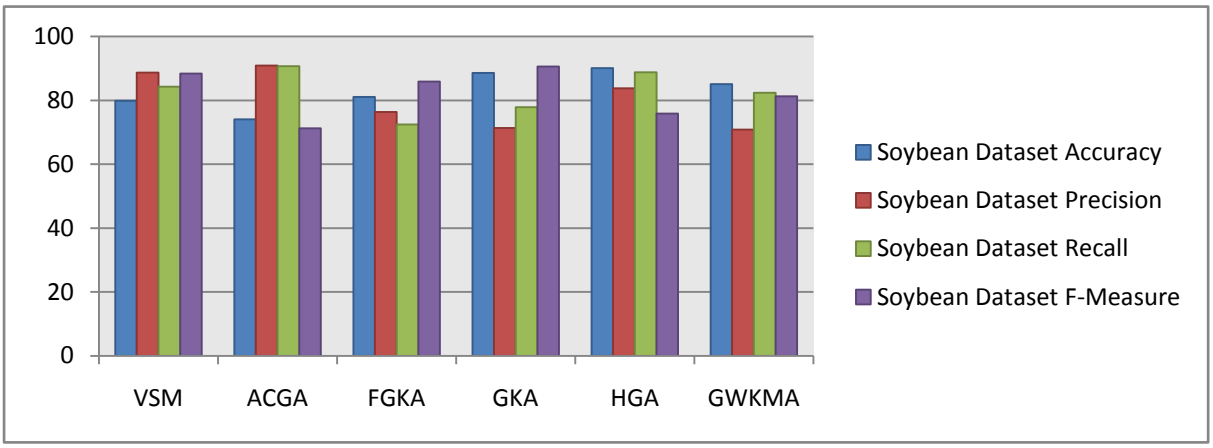
Ionosphere Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
VSM	79.45	88.91	84.77	88.89
ACGA	74.91	90.08	90.78	70.56
FGKA	80.98	76.67	72.89	85.78
GKA	88.67	70.67	77.78	90.67
HGA	90.56	83.45	88.34	75.89
GWKMA	72.12	73.9	80.67	75.66



The above graph shows that performance of Ionosphere dataset. The Accuracy of HGA algorithm is 90.56 which is higher when compare to other five (VSM, ACGA, FGKA, GKA, GWKMA) algorithms. The Precision of ACGA algorithm is 90.08 which is higher when compare to other five (VSM, HGA, FGKA, GKA, GWKMA) algorithms. The Recall of ACGA algorithm is 90.78 which is higher when compare to other five (VSM, HGA, FGKA, GKA, GWKMA) algorithms. The F-Measure of GKA algorithm is 90.67 which is higher when compare to other five (VSM, ACGA, FGKA, HGA, GWKMA) algorithms.

VIII d. SOYBEAN DATASET RESULTS

Soybean Dataset				
Algorithm	Accuracy	Precision	Recall	F-Measure
VSM	79.89	88.65	84.23	88.34
ACGA	74.03	90.89	90.67	71.23
FGKA	81.08	76.32	72.45	85.9
GKA	88.54	71.32	77.89	90.56
HGA	90.08	83.78	88.78	75.9
GWKMA	85.09	70.89	82.33	81.23



The above graph shows that performance of Soybean dataset. The Accuracy of HGA algorithm is 90.08 which is higher when compare to other five (VSM, ACGA, FGKA, GKA, GWKMA) algorithms. The Precision of ACGA algorithm is 90.89 which is higher when compare to other five (VSM, HGA, FGKA, GKA, GWKMA) algorithms. The Recall of ACGA algorithm is 90.67 which is higher when compare to other five (VSM, HGA, FGKA, GKA, GWKMA) algorithms. The F-Measure of GKA algorithm is 90.56 which is higher when compare to other five (VSM, ACGA, FGKA, HGA, GWKMA) algorithms.

IX. CONCLUSION

Clustering is an important task with applications in many fields. Heuristic algorithms are used for this task in an attempt to provide acceptable results, both in terms of solution quality and running time, because all of the non-trivial clustering problem variations. GAs has been applied to the clustering problem for many applications with some success as described. For clustering on very large data sets, such as image data sets, the size of the associated databases makes it necessary to modify traditional GAs because of their slow running times [4]. In this paper we proposed a steady GA algorithm with efficient encoding technique and GA operators along with input set preprocessing. Our GA provided better quality solutions faster than the k-means algorithm.This paper has presented the idea of using hierarchical agglomerative clustering on a bilingual parallel corpus [6]. The aim has been to illustrate this technique and provide mathematical measures, which can be utilized to quantify the similarity between the clusters. When we applied the genetic algorithm to the reduced set of terms to tune the weights of the terms (a maximum of 500 terms) to be considered in the clustering process, the result actually showed a drop in the purity of the clusters [8].This constraint makes most of the clustering algorithms, including GA solutions, impractical for this task.

The concept of genetic algorithm can be, however, used to build an agglomerative clustering algorithm. In the proposed Agglomerative Clustering Genetic Algorithm each individual represents one cluster, instead of the whole clustering solution [1]. This allows a pair of individuals to recombine or join the genetic material that is a small subset of the network. The evaluation of newly created individuals can be done based on two clusters and their direct neighbourhood. If ACGA is used to look for clusters of limited size, for example fine-grained

communities in social networks, each of its steps requires limited resources irrespective of the dataset size [3]. The evaluation on datasets generated by two social network models demonstrated that ACGA was generally able to match and often outperform the FGKA algorithm. It was possible even though there is no fair competition between these two approaches. ACGA is based on stricter assumptions [5].An important advantage of the local processing in ACGA is the possibility of distribution of its computation.This study proposed a genetic weighted K-means algorithm (GWKMA) which is a hybrid algorithm of the weighted K-means and a genetic algorithm. GWKMA was run on three real-life datasets. The results of the computational experiments showed that GWKMA can fulfil the clustering tasks [7]. Furthermore, the results also showed that the GWKMA outperformed both the WKMA without GA and other GA-clustering algorithms without the WKMA operator.

X. FUTURE WORK

Thus as future work we can extend it to more algorithms for experiments as a few have been taken for this purpose. The algorithms discussed in this study can be improved further through thorough research made in this field so that its application gets more intense [2]. One of our future work goals is the distribution of the algorithm over the machines of users of a social network. This can assure that the algorithm is not only able to process large networks but it is also scalable as the processing power grows linearly with the number of clustered items [4]. Thus overcomes the disadvantages of the k-means and the weighted k-means. In addition, the proposed algorithm is generic and could have applications to clustering large-scale biological data such as gene expression data and peptide mass spectral data. In future we would like to explore some more objective functions [6]. We would like to test our approach more extensively. In order to select a single solution from the final Pareto optimal front we have developed a semi-supervised approach. In future we would like to develop some more techniques for this purpose [7].

Results reveal that the proposed semi-supervised feature selection technique is capable to detect the appropriate feature combinations and appropriate partitioning from data sets having the point symmetric clusters. Our empirical studies conducted on several real-world datasets confirmed both the effectiveness and efficiency of the proposed algorithm [3]. Finally, the algorithm can also be adapted to work within a inductive framework. In future some other fitness function may be selected or some constraints may be given for selecting clusters. Also based on our observation we can notify that the hybrid approach defined by us is an effective approach in order to retrieve outlier which was performs on the dataset. In future, we can replace clustering model with some other model and compare its classification accuracy with the proposed framework [5]. Our future work includes creation of new methods for classification using supervised projected clustering methods. There is scope for creation of new supervised projected clustering methods using other optimization methods.

REFERENCES

[1] K. Krishna and M. Murty, “Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics”, 29(3):433–439, 2011.

[2] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown, “Fast genetic K-means algorithm and its application in gene expression data analysis”, Technical Report TR-DB-06-2013, http://www.cs.wayne.edu/~luyi/publication/tr0603.pdf, 2013.

[3] J. N. Bhuyan, V. V. Raghavan, and V. K. Elayavalli, “Genetic algorithm for clustering with an ordered representation,” in Proc. 4th Int. Conf. Genetic Algorithms. San Mateo, CA: Morgan Kaufman, 2011.

[4] D. R. Jones and M. A. Beltramo, “Solving partitioning problems with genetic algorithms,” in Proc. 4th Int. Conf. Genetic Algorithms. San Mateo, CA: Morgan Kaufman, 2012.

[5] G. P. Babu and M. N. Murty, “A near-optimal initial seed selection in K-means algorithm using a genetic algorithm,” Pattern Recognit. Lett, vol. 14, pp. 763–769, 2013

[6] S. Bandyopadhyay and U. Maulik, “Nonparametric genetic clustering: Comparison of validity indices”, IEEE Trans. Syst., Man, and Cybern. C, Appl. Rev., vol. 31, no. 1, pp. 120–125,2009.

[7] S. Guha, R. Rastogi, and K. Shim," ROCK: A robust clustering algorithm for categorical Attributes", Information System, vol. 25, no. 5, pp. 345–366, 2010.

[8] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown, "Incremental genetic K-means algorithm and its application in gene expression data analysis", BMC Bioinformatics 5:172, 2011.

[9] M. Mitchell, "An Introduction to Genetic Algorithms", MIT Press, 2009.

[10] M. Painho and F. Bação, "Using genetic algorithms in clustering problems," in Proceedings of Geo Computation Conference, 2010.

[11] S. Basu, M. Bilenko, and R.J. Mooney, "A probabilistic framework for semi-supervised clustering", In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 56–68, 2014.

[12] Yin Xiang, Alex Tay Leng Phuan," Genetic Algorithm Based K-Means Fast Learning Artificial Neural Network", Nanyang Technological University, 2014.

[13] Y. Liu, Kefe and X. Liz," A Hybrid Genetic Based Clustering Algorithm", Proceedings of the Third International Conference on Machine Leaming and Cybernetics, Shanghai, 26-29 August 2012.