Chapter 3

Optimizer augmented clustering (OAC)

3.1 Why Optimization in Clustering?

The diagram highlights the utility of optimization in the context of clustering. If one need to cluster the set of given points depicted in the figure, what should be the right number of clusters? This is a subjective issue, and as witnessed in the diagram above, should have ideally stopped when the number of clusters is three (3), as increasing the number of clusters (4 or 5) tends to make the clusters look rather sparse. In algorithms like K-Means, the user has to specify the number of clusters k.

3.2 Focus of Investigation  The central thrust of the dissertation is to investigate the benefits of employing heuristic and other evolutionary based optimization techniques such as genetic algorithm,
particle swarm optimization, differential evolution, simulated annealing among many others on free parameters of clustering. We propose a multi-level method employing an optimization loop, in addition to the clustering algorithm, to achieve higher clustering performance through an automatic, better choice of these free parameters.

As a case study, we have studied it on the benefits obtained for the iterative centroid improvement of K-Means, and thereafter on another clustering algorithm Maximin to estimate the recurrence of the improved results obtained using K-Means. Some initial experiments have been performed in the area of text clustering. For the document source, we used 1060 records created for journal articles from the Information Science and Abstracts (ISA) database and computer science technical reports collected from various sites on the Internet. This is a comprehensive subset of the entire database. The dataset ISA used for experimentation contains approximately 5,000 documents culled from abstracts of ACM. We have experimented using a subset of 1060 documents (referred as Bigcheck) which is a fairly representative subset of the entire dataset. Each document record includes a complete abstract, title, author, and subject keywords.

After the initial experiments conducted in the domain of textual data sets, we have later extended our experiments to standard dataset SOYBEAN and WATER TREATMENT PLANT. Four optimizing techniques viz., GA, SA, DE and PSO were used to arrive at better centroids compared with an independent K-Means and Maximin. The performances of these methods and an independent K-Means (& later with Maximin) run were compared and studied using metrics for compactness of clusters. To assess the improvement in cluster quality of the methods, we use two quantitative measure Silhouette coefficient and Davies Bouldin Index.

### 3.3 Optimizer Augmented Clustering (OAC)

Section 3.2 contains an introduction to our investigation. In this section, we represent pictorially the basic OAC scheme and the proposed work. We provide the architecture of Optimizer Augmented Clustering (OAC). The basic schematic of the OAC technique is shown in Fig. 3.2. Algorithm 3.1 depicts K-Mean algorithm vis-à-vis OAC.

---

As can be observed from Table 2.3, a comprehensive systematic study of recent optimization based methods (evolutionary and heuristic based) on free parameters of some important clustering algorithms (like K-means & Maximin) can reveal useful information about performance related issues of such optimizer augmented algorithms. This is the prime objective of this investigation.
Fig 3.2: Basic schematic of OAC

3.4 Optimization Approaches Used For Experimentation

3.4.1 Genetic Algorithms:
By using exhaustive search techniques, we can solve small problems to optimality, although the time complexity may be enormous. For certain applications, it may well pay to spend extra time to be certain of the optimal solution. The use of Genetic Algorithms (GA) in improving the K-means clustering process has already been reported. 27 28. 27 exploits the searching capability of genetic algorithms in order to search for appropriate cluster centers in the feature space such that a similarity metric of the resulting clusters (within cluster spread) is optimized. 28 present a genetic algorithm for selecting centers to seed the
Algorithm 3.1  
K-Mean algorithm vis-à-vis OAC

The K-Mean algorithm's major steps can be summarized as follows:  
Step 1 - begin with an arbitrary assignment of samples to the clusters;  
Step 2 - compute the sample mean of each cluster;  
Step 3 - reassign each sample according to the nearest mean;  
Step 4 - if the classification of all samples has not changed, stop; else go to step 2.

The OAC algorithm can be summarized as follows:  
Step 1 – begin with an arbitrary assignment of initial population  
Step 2- compute fitness values of individuals  
Step 3- Use fitness (maximize) to generate candidate solution/ improve solution, forming a new set of population  
Step 4- If termination criteria is met, stop; else go to step 2

Though essentially our work can be considered as similar with Laszlo et al. in that the objective of investigation is the same i.e., finding good centers for K-means, yet the approach adopted is vastly different. With reference to 28 'A genetic algorithm that exchanges neighboring centers for k-means clustering', the following difference was observed:

1. Laszlo et al. uses a novel region based crossover operator that exchanges neighboring centers. However, in our crossover approach, we’re using a random 5-point crossover approach.

2. Laszlo et al. have used a crossover operator that exchanges subsets of centers that occupy the same region of space. Whereas our crossover operator exchanges random subsets of centers between parents.
3. As their GA proceeds, Laszlo et al. evolves individuals with different no. of genes. In our case, the no of genes in individual evolution remains the same.

4. For selection, Laszlo et al. uses a roulette-wheel sampling with replacement, whereas we have used steady state replacement based on mating pool and higher fit values for selection.

5. Laszlo et al. have used mutation with a low probability, whereas we have not used mutation.

6. The work of Laszlo et al. is inspired by a data structure oriented approach, being based on their earlier work employing hyper-quadtrees which they claim gave good results for low dimensional data.

We are using GA to arrive at better selection of centroids in the inner loop. Thereafter, the K-Means algorithm is applied and clustering is performed. In our method, we have used a random 5-point crossover approach, exchanging random subsets of centers between parents for crossover, and keeping the number of genes in individual evolution the same. We have used steady state replacement based on 20% mating pool and higher fit values for selection. Specific to our work on using Steady GA, it may be mentioned that our work is vastly different from the commonly cited works on using GA to clustering using K-Means. Some of these, employed GA for clustering, and instead of crossover used K-Means as a genetic operator for clustering. A few other approaches used a novel region based crossover operator that exchanges neighbouring centers, evolving individuals with different number of genes using a roulette-wheel sampling with replacement, and also use mutation with a low probability.

3.4.2 Particle Swarm Optimization (PSO):

PSO is another computational intelligence method that we have used. It has already been applied to image clustering and other low dimensional datasets [V. D. Merwe et al., 2003][M. Omran et al., 2002]. However, to the best of the author’s knowledge, PSO has not been used to cluster text documents [Xiaohui Cui et al., 2005]. In this investigation, in the initial experiments, the corpus of documents have been reduced to an equivalent numeric dataset on which PSO augmented K-Means clustering algorithm have been implemented and evaluated for performance. Compared with genetic algorithms (GAs), the information sharing mechanism in PSO is significantly different. In GAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only gBest (or lBest) gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases.
In our PSO optimizer, we have considered the set of centroids (fifteen for \(k=15\)) as a particle. A collection of fifty such particles, the same set that was used during the experimentation with GA, forms the collective population or the swarm. During initialization of the parameters, we set the max iteration to 50 to find the particle best position. Each iteration updates the particle position and its velocity. Thereafter, the particle best and global best values are computed. The fitness value of the swarm is computed for successive iterations and compared. We terminate the procedure when the fitness value doesn't improve over five generations / iterations.

Contrary to the localized searching of the K-means algorithm, the PSO clustering algorithm performs a globalized search in the entire solution space. In the experiments we conducted, we applied the PSO in the inner loop before application of the K-Means clustering algorithm to obtain better centroids for iterative K-Means. The results highlight that the hybrid PSO algorithm can generate more compact clustering results than the K-Means algorithm.

3.4.3 Simulated Annealing:

The third optimizer we used, Simulated Annealing (SA) optimizer, uses an iterative procedure. Each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter \(T\) (called the temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when \(T\) is large, but increasingly "downhill" as \(T\) goes to zero. The allowance for "uphill" moves saves the method from becoming stuck at local minima— which are the bane of greedier methods. The perturbation operator used in general annealing has a simple meaning in clustering: it amounts to a relocation of a point from its current to a new randomly chosen cluster (very similar to \(k\)-means scheme).

In our experiments, the Maximum number of iterations before the cooling schedule is applied has been set to 10. The global parameter \(T\) have been set to 100, and thereafter a cooling schedule of \(t_i = t_i \times 0.95\) was used. The ‘uphill’ (bad) solutions were accepted with a probability of 0.9 to overcome local convergence. The convergence criteria taken was \(T=0\). Since a particular iteration required a number of cooling schedules, taking quite a good amount of time with each cooling schedule, fitness function was computed after clustering with values obtained after each cooling schedule.
3.4.4 Differential Evolution:

The fourth optimizer, Differential Evolution, has become very popular recently as an effective method for global optimization problems. In this technique, a new trial point \( y \) is generated by using mutation and crossover. There are various strategies on how to create the mutant point \( u \). One of the most popular strategy called DE/rand/1/ generates the point \( u \) by adding the weighted difference of two points

\[
\begin{align*}
u = r_1 + F (r_2 - r_3)
\end{align*}
\]

where \( r_1, r_2 \) and \( r_3 \) are three mutually distinct points taken randomly from population \( P \), not coinciding with the current \( x_i \) and \( F > 0 \) is an input parameter. Another strategy called DE/best/2/ generates the point \( u \) according to formula

\[
\begin{align*}
u = x_{\text{mm}} + F (r_1 + r_2 - r_3 - r_4)
\end{align*}
\]

where \( x_{\text{mm}} \) is the point of \( P \) with minimal function value, \( r_1, r_2, r_3, r_4 \) are four mutually distinct points taken randomly from \( P \) not coinciding with the current \( x_i \) or \( x_{\text{mm}} \) and \( F > 0 \) is an input parameter. In our experiments, we have used DE/best/2/ method. The efficiency of differential evolution is very sensitive to the setting of values \( F \). The values recommended in literature are \( F = 0.8 \) and we have used this value in our experiments.

3.4.5 Cluster Quality:

To assess the improvement in Cluster Quality from the use of these optimizer methods, we use a quantitative measure Silhouette coefficient (SC). Three datasets were used for the experimentation viz., synthetic text dataset (ISA dataset, named as BIGCHECK & used in the earlier experiment), and two standard datasets SOYBEAN & WATER TREATMENT PLANT. In addition to the earlier
four methods, two more flavors of GA methods were used: Roulette Method & Grouping GA (GGA). The number of instances in SOYBEAN dataset is 307, and the number of attributes is 35. The number of instances in WATER TREATMENT PLANT dataset is 527, and the number of attributes is 38. The Silhouette Coefficient values of the resulting clusters were computed for each of these three datasets (ISA, SOYBEAN & WATER TREATMENT PLANT) for each of the following seven methods: K-Means, GA Steady State K-Means, GA Roulette Wheel K-Means, Grouping GA K-Means, PSO K-Means, SA K-Means & DE K-Means.

The results obtained were further verified using another quantitative measure, the popular Davies Bouldin index. The same three datasets (BIGCHECK, SOYBEAN & WATER TREATMENT PLANT) and the seven methods (K-Means, GA Steady State K-Means, GA Roulette Wheel K-Means, Grouping GA K-Means, PSO K-Means, SA K-Means & DE K-Means) were used for the experiments. The experimental results are discussed in the next chapter.