Multiple Cooperative Swarms for Data Clustering

by

Abbas Ahmadi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abbas Ahmadi

Abstract

Exploring a set of unlabeled data to extract the similar clusters, known as data clustering, is an appealing problem in machine learning. In other words, data clustering organizes the underlying data into different groups using a notion of similarity between patterns.

A new approach to solve the data clustering problem based on multiple cooperative swarms is introduced. The proposed approach is inspired by the social swarming behavior of biological bird flocks which search for food situated in several places. The proposed approach is composed of two main phases, namely, initialization and exploitation. In the initialization phase, the aim is to distribute the search space among several swarms. That is, a part of the search space is assigned to each swarm in this phase. In the exploitation phase, each swarm searches for the center of its associated cluster while cooperating with other swarms. The search proceeds to converge to a near-optimal solution. As compared to the single swarm clustering approach, the proposed multiple cooperative swarms provide better solutions in terms of fitness function measure for the cluster centers, as the dimensionality of data and number of clusters increase.

The multiple cooperative swarms clustering approach assumes that the number of clusters is known a priori. The notion of stability analysis is proposed to extract the number of clusters for the underlying data using multiple cooperative swarms. The mathematical explanations demonstrating why the proposed approach leads to more stable and robust results than those of the single swarm clustering are also provided.

Application of the proposed multiple cooperative swarms clustering is considered for one of the most challenging problems in speech recognition: phoneme recognition. The proposed approach is used to decompose the recognition task into a number of subtasks or modules. Each module involves a set of similar phonemes known as a phoneme family. Basically, the goal is to obtain the best solution for phoneme families using the proposed multiple cooperative swarms clustering. The experiments using the standard TIMIT corpus indicate that using the proposed clustering approach boosts the accuracy of the modular approach for phoneme recognition considerably.

Acknowledgements

In the name of God, I would like to thank all the people who helped me during my PhD journey. First of all, I would like to express my sincere gratitude to my supervisors, Professor Fakhri Karray and Professor Mohamed Kamel, for their trust on me from my early days here in Waterloo, for their inspiring guidance and intellectual influence during my research and for their great support. They were not only supportive and helpful supervisors, but knowledgable mentors. My thanks also go to my comprehensive and defense committee members: Professor Simon Yang, Professor Liang-Liang Xie, Professor Andrew Wong, Professor Kumaraswamy Ponnambalam and Dr. Jiping Sun.

I am grateful to my dear friends in Pattern Analysis and Machine Intelligence (PAMI) Lab. My special thanks go to Mohamed El-Abd, Moataz El Ayadi, Heidi Campbell, Khaled Hamouda, Shady Shehata, Masoud Mahootchi, Masoud Makrehchi and and Sameeh Ullah.

I would like to thank Vicky Lawrence, Systems Design graduate coordinator, for her kind help and support during my PhD program. My dear friend, Professor Shoja Chenouri, was always open to answer my questions. I appreciate his help and kindness.

I owe to my dear father, Adigozal, and my dear mother, Zari, for their continuous supports and patience. I would like to thank my father-in-law, Professor Mohsen Mohebi, for his guidance and unconditional supports.

I can not express with words my sincere appreciation to my dear wife, Azadeh. She made not only a calm and benignant atmosphere at home, but was always helpful in my academic carrier. Her valuable comments and feedbacks have certainly promoted the quality of my work.

At the last stage of my PhD journey, our daughter, Saba, was born. Saba brought us happiness and made our life more meaningful. My mother-in-law, Zohreh, came here the same day as Saba. She stayed with us four months and helped us to relieve the challenges of entering to a new stage of life. Without her devoted help, it was almost impossible to finish this thesis on-time.

Dedication

To my parents

To my wife and daughter

Contents

List of Tables

List of Figures

Abbreviations

PSO	Particle Swarm Optimization
KM	K-Means algorithm
KHM	K-Harmonic Means algorithm
FCM	Fuzzy c-Means algorithm
SOM	Self-Organizing Map
UCI	University of California-Irvine machine learning repository
Comp.	Compactness measure
Sep.	Separation measure
Comb.	Combined measure
TAE	Teaching Assistant Evaluation data set
TIMIT	Texas Instruments/Massachusetts Institute of Technology corpus
KNN	K-Nearest Neighbor
ASR	Automatic Speech Recognition
MFCC	Mel-Frequency Cepstral Coefficient
EM	Expectation-Maximization algorithm
GAs	Genetic Algorithms
GMM	Gaussian Mixture Model

Chapter 1

Introduction

Data clustering is the unsupervised classification of a set of data points into similar groups [1]. As a result, the data points clustered in the same group are more similar to each other than those of other groups.

An example of clustering is provided in Fig. 1.1. In this figure, three different clustering solutions are depicted for the given data points.

Extracting the underlying groups of unlabeled data is an inherently ill-posed problem as compared to the supervised classification. That is, the labels of the data are available in the supervised classification, whereas they are unknown in data clustering.

Data clustering is beneficial for a wide range of applications including, but not limited to, data mining, document retrieval, image segmentation, bioinformatics and speech recognition [1], [2], [3], [4], [5], [6].

Various terms are used in data clustering, which are patterns, clusters, cluster centers, model order, features, similarity measures and cluster validity measures. *Patterns* are a set of observations, data points or feature vectors. The aim is to cluster these patterns into a number of groups. A *cluster* or group is basically composed of a set of similar patterns. Thus, each cluster contains a subset of patterns. Each *cluster center* is associated with a cluster. Having cluster centers known, the labels of the data can be extracted easily. In other words, the solution of a clustering algorithm can be stated by means of cluster centers or the estimated labels. *Model order* indicates the number of clusters for the underlying data. *Features* or attributes are used to represent a pattern. The procedure used to obtain the corresponding features of a pattern is known as a feature extraction. Thus, each pattern is distinguished from others in terms of its features. Further, the number of the features indicates the dimension of the data. *Similarity measures* are used to assess the proximity of patterns as the goal of clustering is to partition a given set of data into a set of similar groups. Finally, *cluster validity measures* are useful to evaluate the quality of the clustering solutions. A discussion on different similarity and cluster validity measures will be provided in the following chapter.

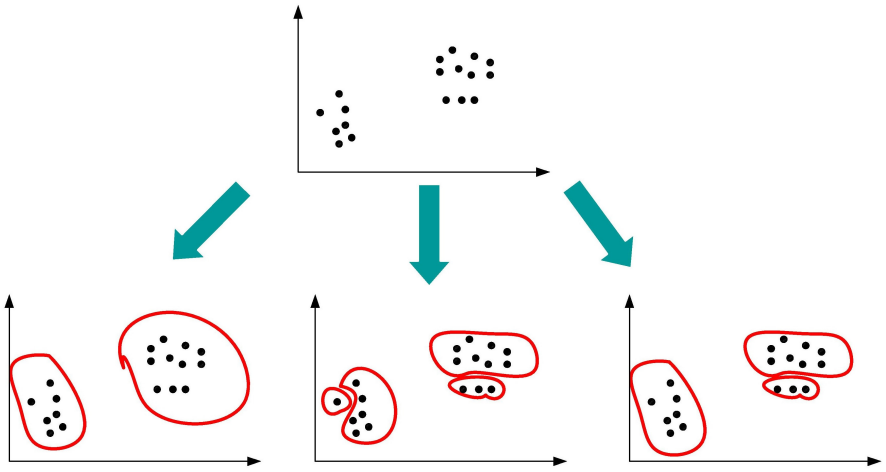


Figure 1.1: Different clustering solutions for the same data

1.1 Motivations

To tackle the clustering problem, several techniques such as K-means, K-harmonic means and fuzzy c-means have been developed. However, most of these techniques either are highly dependent on the initial solutions or are more likely to converge to local optimal solutions. In addition, they do not perform well in manipulating multiple objectives.

Swarm intelligence has recently attracted a great deal of interest from researchers of different backgrounds. Swarm intelligence was biologically inspired, by studying the swarming behaviors of flocks of birds, schools of fish, or swarms of bees [7]. Particle swarm optimization (PSO), as one of the main branches in the swarm intelligence, emulates the social behavior of bird flocks. PSO is a population-based search scheme that tends to find an optimal solution by employing a swarm of individuals referred to as particles. The PSO procedure is less sensitive to the effect of the initial conditions due to its population-based nature. Furthermore, it performs a global search of the solution space. Accordingly, it is more likely to provide a near-optimal solution. Besides, PSO can manage multiple objectives at the same time. As a result, it is an excellent tool for solving clustering problems where optimizing different objectives is of interest.

PSO has been considered to tackle clustering problems [2], [3], [4], [8], [9]. To the best of my knowledge, all of the available approaches use only a single swarm

to deal with the clustering task. However, when the dimensionality of the data is high or the possible number of clusters is large, a single swarm is not sufficient to explore all of the search space. Instead, multiple swarms cooperating together can be considered to obtain cluster centers effectively.

In speech recognition problems, the dimensionality of the data is relatively high and the data contains many clusters. Therefore, the application of multiple swarms clustering has the potential of enhancing the performance on speech recognition problems significantly.

1.2 Goals and Contributions

In this thesis, the aim is to provide an insight into the following issues:

- situating the particle swarm clustering within the taxonomy of the clustering approaches,
- a discussion of clustering using single swarm,
- the formulation of clustering using multiple cooperative swarms,
- model order selection for clustering using multiple cooperative swarms, and
- the application of multiple cooperative swarms clustering for phoneme recognition problem.

The contributions of the thesis can be concisely classified into three main categories. First, a novel clustering approach by means of multiple cooperative swarms is proposed. Then, a stability-based approach is suggested to extract the model order of underlying data using the multiple cooperative swarms approach. Finally, the application of the proposed clustering approach is studied to tackle phoneme recognition. The first two categories are basically considered to demonstrate the theoretical aspect of the thesis. The third category is intended to provide a successful example for the application of the proposed approach.

1.3 Organization

After a brief introduction and discussion of the motivations, goals and contributions, the organization of the thesis will be as follows: In the following chapter, we provide an overview on data clustering. First, hierarchical and partitional approaches for data clustering are outlined. An introduction to single swarm clustering is then provided. Similarity and cluster validity measures are next described.

In chapter three, a detailed explanation of the multiple cooperative swarms clustering approach is presented. Moreover, the performance of the proposed clustering approach as compared to the other clustering approaches is examined.

In chapter four, the stability-based scheme to estimate the model order of the underlying data using multiple cooperative swarms clustering is presented. This technique enables the multiple cooperative swarms clustering to extract the number of the clusters as well. Furthermore, the proposed approach is evaluated using different data sets and its performance is compared with that of other clustering techniques.

In chapter five, the application of the proposed multiple cooperative swarms approach for phoneme recognition is presented. The proposed approach is applied to divide the phoneme recognition task into different subtasks in a modular-based classifier.

In chapter six, conclusions are drawn and future research directions are proposed.

Chapter 2

Overview on Data Clustering

The process of extracting similar groups of the underlying patterns is referred to as data clustering, which is a difficult problem combinatorially [1]. To tackle clustering problem, two major approaches are available. In this chapter, these approaches are outlined. Similarity measures and cluster validity measures are also studied.

2.1 Introduction

Clustering is the process of separating data Y of dimension d into a number of groups $C^{(k)},\ k = 1, 2, \cdots, K$, based on some similarity measures [10]. As a result, each cluster $C^{(k)}$ contains a set of similar data points given by $C^{(k)} = \{\mathbf{y}_j^{(k)}\}_{j=1}^{n_k}$, where $\mathbf{y}_j^{(k)}$ denotes data point j in cluster k and n_k indicates the total number of data points in cluster k. The union of all clusters forms Y:

$$Y = \cup_{k=1}^{K} C^{(k)}, \tag{2.1}$$

where K is the number of clusters. In hard clustering, the clusters are pair-wise disjoint; i.e.,

$$C^{(k)} \cap C^{(k')} = \emptyset, k \neq k' \in [1, ..., K]. \tag{2.2}$$

In other words, each sample or data point from Y is assigned to only one cluster. Let $A_K(Y)$ denote a clustering algorithm aiming to cluster data set Y to K distinct clusters. Moreover, assume the solution of a clustering algorithm $A_K(Y)$ for the given data points Y of size N is presented by $T := A_K(Y)$ which is a vector of labels $T = \{\mathbf{t}_i\}_{i=1}^{N}$, where $t_i \in \mathcal{L} := \{1, ..., K\}$.

To deal with the clustering task, there are two main approaches, namely hierarchical and partitional clustering [1].

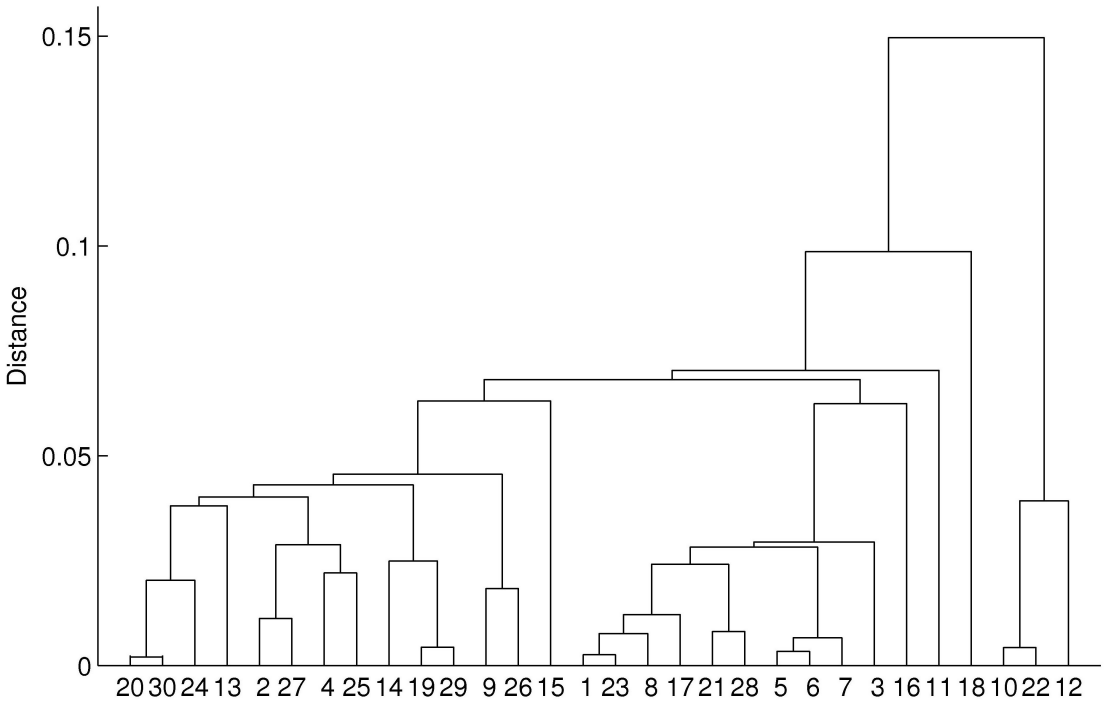


Figure 2.1: An example of the dendrogram

2.2 Hierarchical Clustering Approaches

Hierarchical clustering approaches yield a hierarchy of clusters represented in a tree format referred to as a dendrogram. The dendrogram provides a nested grouping of data points [1]. An example of the dendrogram for a data set of 30 points is shown is Fig. 2.1.

To build the dendrogram, agglomerative and divisive approaches are used. A divisive approach begins with a single cluster containing all data points. It then divides this cluster into two separate clusters. This procedure continues until each cluster includes a single data point. In contrast to the divisive approach, an agglomerative approach considers each data point as a cluster at the beginning. Then, the two close clusters merge together and make a new cluster. Merging close clusters is continued until all points form a single cluster.

Assume clusters $C^{(1)}, \cdots, C^{(k1)}$ are given and the aim is to determine the closest two clusters. Two closest clusters are obtained by

$$(i^*, j^*) = arg \min_{(i,j)} \{d(C^{(i)}, C^{(j)}) | i, j = 1, \cdots, k1, i \neq j\}, \tag{2.3}$$

where $d(C^{(i)}, C^{(j)})$ indicates the distance between two clusters $C^{(i)}$ and $C^{(j)}$. Different notions are available to measure the distance between two clusters, which are single linkage, complete linkage and average linkage [1].

- Single linkage: The minimum distance between all pairs of data points, one from each cluster, is given by

$$d_{single}(C^{(1)}, C^{(2)}) = \min\{d(\nu, \omega) | \nu \in C^{(1)}, \omega \in C^{(2)}\}. \tag{2.4}$$

- Complete linkage: The maximum distance between all pairs of data points, one from each cluster, is defined as

$$d_{complete}(C^{(1)}, C^{(2)}) = \max\{d(\nu, \omega) | \nu \in C^{(1)}, \omega \in C^{(2)}\}. \tag{2.5}$$

- Average linkage: The mean distance between all pairs of data points, one from each cluster, is computed by

$$d_{average}(C^{(1)}, C^{(2)}) = \frac{1}{n_1 \cdot n_2} \sum_{\nu \in C^{(1)}} \sum_{\omega \in C^{(2)}} d(\nu, \omega). \tag{2.6}$$

Different notions of distance between clusters in 2D space are shown in Fig. 2.2. As can be seen, there are two clusters (cluster 1 and cluster 2), each of which is represented by a set of points connected by solid lines, and there is a single point (cluster 3) whose distances from the clusters 1 and 2 are shown in dashed lines. In panels (a) and (b), the shortest dashed-line indicates the closest cluster with which the individual point (cluster 3) is merged. In panel (c), the dashed lines show the distance from different data points. To determine the closest cluster, the mean distance from each cluster is obtained. The shortest mean distance corresponds to the closest cluster.

2.3 Partitional Clustering Approaches

Partitional clustering algorithms divide the data set into a specified number of clusters. The division of the given data points into a set of clusters is done by optimizing a certain criterion [2]. This section concentrates on introducing a number of partitional clustering approaches such as K-means, K-harmonic means, fuzzy c-means and evolutionary-based clustering algorithms.

2.3.1 K-means algorithm

K-means algorithm is the most popular partitional clustering technique. This algorithm starts from K arbitrary random points as cluster centers denoted by $\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}$. Then, data point $\mathbf{y}_j$ is assigned to cluster k' provided:

$$d(\mathbf{y}_j, \mathbf{m}^{(k')}) \leq d(\mathbf{y}_j, \mathbf{m}^{(k)}), \text{for all } k,\ j, \tag{2.7}$$

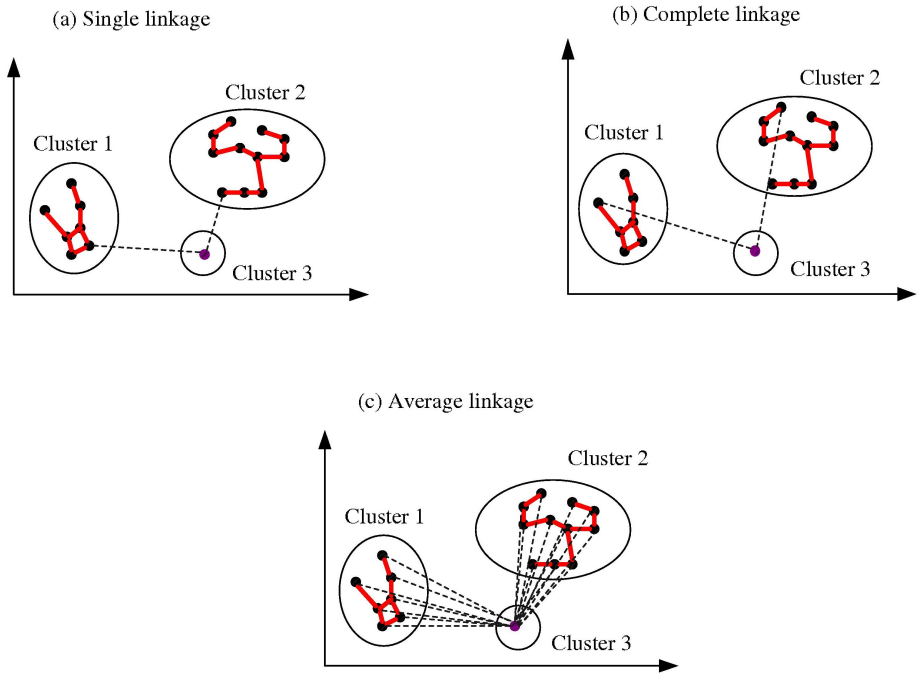


Figure 2.2: Different notions of distance between clusters

where $d(\cdot)$ indicates the Euclidean distance between the two associated points. The centers are next updated according to the corresponding data points as follows:

$$\mathbf{m}^{(k)} = \frac{1}{n_k} \sum_{j=1}^{n_k} \mathbf{y}_j^{(k)}, \ k = 1, \cdots, K \cdot \tag{2.8}$$

This procedure is repeated till the termination criterion such as maximum number of iterations or number of iterations with no improvement is attained. Having a set of data Y, the procedure of the k-mean clustering is presented in Algorithm **2.1**.

Algorithm 2.1 K-means clustering algorithm

1: Pick K either arbitrary prototypes or from training data and denote them $\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}$.
2: For each $\mathbf{y}_j$, determine the nearest prototype. Assign it to cluster k' if $d(\mathbf{y}_j, \mathbf{m}^{(k')}) \leq d(\mathbf{y}_j, \mathbf{m}^{(k)})$, for all k.
3: Update each prototype as follows: $\mathbf{m}^{(k)}$= sample mean of all data points assigned to cluster k.
4: If no prototype changed in step 3 from its previous value, stop. Otherwise, go back to step 2.

As can be seen in K-means procedure, it tries to minimize the following objective

function:

$$F_{KM} = \sum_{k=1}^{K} \sum_{j=1}^{n_k} d^2(\mathbf{y}_j^{(k)}, \mathbf{m}^{(k)}). \tag{2.9}$$

This objective function is designed in such a way as to yield as compact clusters as possible [1]. The K-mean clustering has many features that make it a popular approach. It converges to a solution very quickly and it is easy to understand and implement. However, there are some issues with this algorithm as well. The K-means algorithm is highly sensitive to initial solutions and it may converge to local optimal solutions.

2.3.2 K-harmonic means algorithm

Zhang et al. have introduced a new method known as K-harmonic means (KHM), which uses the harmonic averages of distances from every data point to the centers. They showed empirically that their method is less sensitive to initial solutions. As compared to K-means, KHM improves the quality of the clustering results in certain cases [11].

The harmonic average of N points $\{y_j\}_{j=1}^N$ is defined as

$$HA(\{y_j\}_{j=1}^N) = \frac{N}{\sum_{j=1}^N \frac{1}{y_j}}. \tag{2.10}$$

The required objective function of the K-harmonic means algorithm is also given by

$$F_{KHM} = \sum_{j=1}^{N} \frac{K}{\sum_{k=1}^K \frac{1}{||\mathbf{y}_j - \mathbf{m}^{(k)}||^2}}, \tag{2.11}$$

where $||\cdot||$ stands for the Euclidean norm. By taking the partial derivatives of the F_{KHM} with respect to centers, $\mathbf{m}^{(k)}$, $k = 1, \cdots K$, and setting them to zero, the recursive updating rule of centers is obtained as

$$\mathbf{m}^{(k)} = \frac{\sum_{j=1}^N \frac{1}{d_{j,k}^3 (\sum_{k=1}^K \frac{1}{d_{j,k}^2})^2} \mathbf{y}_j}{\sum_{j=1}^N \frac{1}{d_{j,k}^3 (\sum_{k=1}^K \frac{1}{d_{j,k}^2})^2}}, \tag{2.12}$$

where $d_{j,k} = ||\mathbf{y}_j - \mathbf{m}^{(k)}||$. By starting from an initial solution for centers, this recursive procedure is continued to converge to final clustering centers [11].

It is not obvious how to interpret the objective function.

2.3.3 Fuzzy c-means algorithm

Bezdeck has proposed an extension for K-means using fuzzy logic named fuzzy c-means (FCM) clustering. In FCM, every data point is associated to each cluster with some degree of membership [12]. That is, it has a membership in all clusters. In FCM clustering, it is desired to minimize the following objective function:

$$F_{FCM} = \sum_{k=1}^{K} \sum_{j=1}^{N} \mu_{j,k}^{\rho} \cdot ||\mathbf{y}_j - \mathbf{m}^{(k)}||^2, \ 1 \leq \rho \leq \infty, \tag{2.13}$$

where ρ is a fuzziness parameter and $\mu_{j,k}$ shows the degree of membership $\mathbf{y}_j$ in cluster k given by

$$\mu_{j,k} = \frac{1}{\sum_{k'=1}^{K} (\frac{||\mathbf{y}_j - \mathbf{m}^{(k)}||}{||\mathbf{y}_j - \mathbf{m}^{(k')}||})^{\frac{2}{\rho-1}}}. \tag{2.14}$$

Furthermore, the new centers, $\mathbf{m}^{(k)}, k = 1, \cdots, K$, in fuzzy c-means clustering are modified as

$$\mathbf{m}^{(k)} = \frac{\sum_{j=1}^{N} \mu_{j,k}^{\rho} \cdot \mathbf{y}_j}{\sum_{j=1}^{N} \mu_{j,k}^{\rho}}, \qquad k = 1, \cdots, K. \tag{2.15}$$

The required procedure for fuzzy c-means clustering is presented in Algorithm **2.2**.

Algorithm 2.2 Fuzzy c-means clustering algorithm

1: Pick K prototypes, either arbitrary or from training data and denote them $\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}$.

2: Compute $\mu_{j,k}$ for all j and k:
$\mu_{j,k}(t+1) = \frac{1}{\sum_{k'=1}^{K} (\frac{||\mathbf{y}_j - \mathbf{m}^{(k)}(t)||}{||\mathbf{y}_j - \mathbf{m}^{(k')}(t)||})^{\frac{2}{\rho-1}}}$, for $||\mathbf{y}_j - \mathbf{m}^{(k')}(t)|| > 0, \ \forall \ j, k$.

3: Update each prototype: $\mathbf{m}^{(k)}(t+1) = \frac{\sum_{j=1}^{N} \mu_{j,k}^{\rho}(t+1) \cdot \mathbf{y}_j}{\sum_{j=1}^{N} \mu_{j,k}^{\rho}(t+1)}$.

4: If no prototype changed in step 3 from its previous value, stop. Otherwise, $t = t + 1$ and go back to step 2.

2.3.4 Evolutionary-based clustering algorithms

Finally, evolutionary-based clustering techniques are based on evolutionary approaches. Swarm-based clustering approaches, inspired by the swarming behavior of living beings such as ants, bees and flocks in nature [13], are examples of evolutionary-based clustering. There are two main swarm-based approaches, which are ant-based and PSO-based clustering. A comprehensive study on the latest approach will be provided in the following section. In ant-based clustering, each artificial ant picks up and drops down items on the basis of probabilistic behavior [13].

In the ant-based clustering, the pioneering work has been done by Deneubourg et al. [14] in which artificial ants move randomly on a square grid of cells containing some items. Whenever an unloaded ant is faced with an item located in a cell, it picks up the item with some probability depending on the density of the similar items in the surrounding region. Subsequently, when a loaded ant encounters a free cell, it drops the item with some probability. Eventually, all the similar items are classified in the same cluster.

An extension to the previous work has been suggested by Lumber and Faieta [15] using a dissimilarity type of evaluation for the local density. Also, Monmarche [16] has proposed a new algorithm in which it is possible to have several items in one cell. Moreover, some researchers have investigated the ant-based clustering in combination with fuzzy c-means and fuzzy rules [17].

2.4 Particle Swarm Clustering

Before outlining the particle swarm clustering algorithm, a description of the particle swarm optimization (PSO) is provided.

2.4.1 PSO Procedure

PSO as a search technique was mainly introduced to tackle optimization problems [18], [19], [20]. The PSO procedure commences with an initial swarm of particles and evolves through a number of iterations to find an optimal solution given a predefined fitness function f. Each particle is characterized by a position-vector $\mathbf{x}_i$ and velocity-vector $\mathbf{v}_i$· Each particle contains a vector which keeps track of the best position that it was situated at for any iteration. This variable is referred to as the particle's personal best, denoted by $\mathbf{x}_i^{pb}$. The swarm also keeps track of the best position that has been found by all particles, i.e. the best position of all the personal best positions. This variable is called global best, denoted by $\mathbf{x}^*$. A new velocity and position of each particle for time step $t+1$ is obtained by the use of the following equations

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{x}_i^{pb}(t) - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{x}^*(t) - \mathbf{x}_i(t)), \tag{2.16}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1), \tag{2.17}$$

where w is inertia weight to control the impact of the previous history of velocities on the current one, c_1 and c_2 are positive constants known as cognitive and social components, respectively, and, r_1 and r_2 are samples of random variables uniformly distributed in the interval $[0,1]$; i.e., $r_1, r_2 \sim U(0,1)$. As can be seen from the above-mentioned equations, to produce a new position, each particle follows two

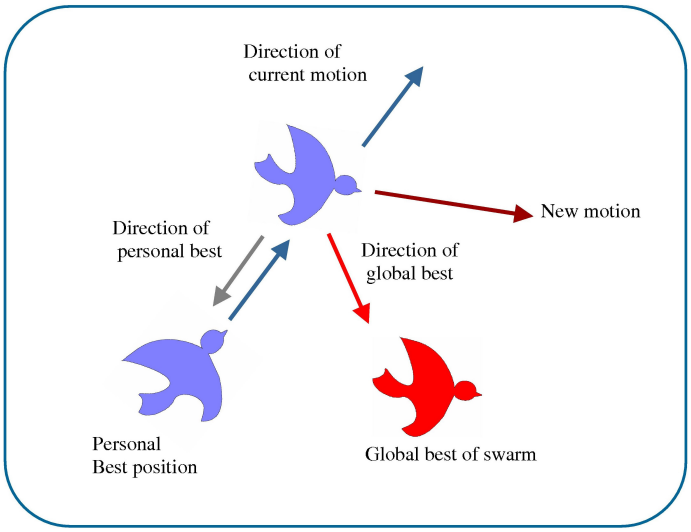


Figure 2.3: Schematic presentation of updating the velocity of a particle

best values, which are the personal best and global best of the swarm obtained so far. The schematic representation of updating the velocity of a particle is illustrated in Fig. 2.3. For this specific example, the personal best of the given particle is situated at its previous position.

For minimizing the fitness function, the personal best position of particle i at time step t is updated as follows:

$$\mathbf{x}_i^{pb}(t+1) = \begin{cases} \mathbf{x}_i^{pb}(t) & \text{if} \quad f(\mathbf{x}_i(t+1)) \geq f(\mathbf{x}_i^{pb}(t)), \\ \mathbf{x}_i(t+1) & \text{otherwise.} \end{cases} \tag{2.18}$$

The best particle of the swarm is also updated using the following equation

$$\mathbf{x}^*(t+1) = arg \min_{\mathbf{x}_i^{pb}(t)} \; f(\mathbf{x}_i^{pb}(t)), \; i \in [1, ..., n]. \tag{2.19}$$

The initial velocities can be set to zero

$$\mathbf{v}_i(0) = 0, \; i \in [1, ..., n]. \tag{2.20}$$

Initializing velocities to zero may restrict the search space [21]. Alternatively, one can initiate velocities by generating random values [18], as used in this thesis. To avoid large initial momentum, the initial velocities are set to small values. Large initial velocities lead to large position updates which may cause particles to go away from the defined search space [18].

The personal best for each particle is initialized as

$$\mathbf{x}_i^{pb}(0) = \mathbf{x}_i(0), \quad i \in [1, ..., n]. \tag{2.21}$$

Algorithm 2.3 Pseudocode for PSO procedure

initialize a swarm of size n
repeat
  for each particle $i \in [1, ..., n]$ **do**
    update position and velocity
    if $f(\mathbf{x}_i(t+1)) < f(\mathbf{x}_i^{pb}(t))$ **then**
      $\mathbf{x}_i^{pb}(t+1) \leftarrow \mathbf{x}_i(t+1)$
    end if
  end for
  $\mathbf{x}^*(t+1) \leftarrow arg \min_{\mathbf{x}_i^{pb}(t)} \{f(\mathbf{x}_i^{pb}(t)) |\ i \in [1, ..., n]\}$
until termination criterion is met

There are several methods to terminate a PSO procedure such as reaching the maximum number of iterations, having a number of iterations with no improvement, and reaching minimum objective function criterion [22]. A pseudocode for PSO procedure is provided in Algorithm **2.3**.

Particle swarm optimization has been applied successfully to different classes of optimization problems including constrained optimization, multi-objective optimization, discrete optimization and nonlinear function optimization [18]. Moreover, PSO has been employed to deal with many areas of applied optimization such as neural networks, power systems, image segmentation, bioinformatics, scheduling and data mining [18].

2.4.2 Data Clustering by Means of a Single Swarm

Due to its abilities, PSO has been used in other applications, such as classification and clustering [2], [3], [4], [8], [9], [5], [23], [24].

Xiang et al. have employed a hybrid of PSO and self-organizing map (SOM) to construct a novel scheme for gene clustering [5], [23]. In this method, the weights of the SOM are first trained using competitive learning. The weights are then optimized using PSO. This scheme is called *block SOM/PSO* [5]. They have also proposed another hybrid scheme called *alternating SOM/PSO*. In this scheme, several SOMs are trained over a number of iterations. Then, each SOM is treated as a particle of the associated swarm. These populations of SOMs evolve using PSO over a number of iterations [5].

Cui et al. have proposed a new method for document clustering [9]. In their work, PSO is used to find optimal centers of clusters in the search space based on the average distance of documents from their corresponding centers, which is defined as the fitness function to evaluate the solution provided by each particle. Omran et al. have applied PSO for image clustering [2], [3]. Their proposed method is similar to that of Cui et al., but the main difference is how they define fitness function. They

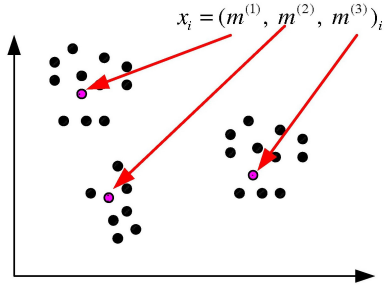


Figure 2.4: Representation of particle's position in single swarm clustering

desire to cluster images such that intra-cluster distance and quantization error are minimized while the distance between clusters is maximized.

Moreover, Cui et al. have introduced another technique for data clustering which combines PSO and K-means clustering methods [4]. This technique, known as *hybrid PSO*, uses PSO to provide the initial seeds for the K-means clustering method.

To apply particle swarm optimization as a clustering technique, one should model the clustering task as an optimization problem. The goal of such a task is to obtain centers of clusters so that an objective function is optimized. In other words, the definition of cluster center depends on the definition of the objective function being optimized by PSO algorithm. If the objective function is represented by means of the compactness measure, the centers are the same as cluster means. However, if the objective function is formulated by some other cluster validity measure such as separation measure, the cluster centers are not necessarily the same as cluster means.

Assume Y is a set of data points intended to be clustered into K separate clusters. Therefore,

$$Y = \cup_k^K C^{(k)}. \tag{2.22}$$

Also, assume n_k is the number of data samples in cluster k. Moreover, each particle is represented as $\mathbf{x}_i = (\mathbf{m}^{(1)}, ..., \mathbf{m}^{(K)})_i$, where $\mathbf{m}^{(k)}$ denotes the center of cluster k. For the sake of simplicity, the representation of particle i is hereafter denoted by $\mathbf{x}_i = \mathbf{M}_i$, where $\mathbf{M} = (\mathbf{m}^{(1)}, ..., \mathbf{m}^{(K)})$. In other words, each particle contains a representative for the center of all clusters. The representation of particle's position, $\mathbf{x}_i$, for $K = 3$ clusters is illustrated in Fig. 2.4.

To model the clustering problem as an optimization problem, it is required to define constraints as well as an objective function. The only constraint is that the points should be selected from the domain of the data set or search space. The objective function can be modeled by means of the cluster validity measures, such as compactness and separation.

After defining the fitness function in terms of a cluster validity measure, a single swarm can be used to obtain the solution of the clustering problem. The search

starts from an initial population in the solution space and proceeds to find a near-optimal solution.

The required pseudocode for single swarm clustering is presented in Algorithm **2.4**.

Algorithm 2.4 Pseudocode for single swarm clustering

```
initialize a swarm of size n
repeat
  for each particle i ∈ [1, ..., n] do
    update position and velocity
    if F(M_i(t+1)) < F(M_i^pb(t)) then
      M_i^pb(t+1) ← M_i(t+1)
    end if
  end for
  M*(t+1) ← argmin{F(M_i^pb(t)) | i ∈ [1, ..., n]}
until termination criterion is met
```

When the dimensionality of the data is high and the number of clusters is large, the ability of the single swarm clustering is not sufficient to probe all of the search space. Instead, multiple cooperative particle swarms can be considered to determine clusters' centers as explained in chapter 3.

2.5 Similarity Measures

As mentioned earlier, similarity measures play an important role in identifying different groups of the underlying data [1]. The notion of similarity between data points is usually represented using their corresponding distance. The smaller the distance is, the more similar the points will be. Several functions are available to measure the distance between two data points $\mathbf{y}_1$ and $\mathbf{y}_2$.

- Minkowski distance: This measure is defined as

$$d_{Minkowski}(\mathbf{y}_1, \mathbf{y}_2) = (\sum_{i=1}^{d} |\mathbf{y}_{1,i} - \mathbf{y}_{2,i}|^p)^{\frac{1}{p}}. \tag{2.23}$$

  By setting $p = 1$ and $p = 2$, Manhattan distance and Euclidean distance are obtained, respectively, as special cases of Minkowski distance. In Fig. 2.5, a graphical illustration of Manhattan and Euclidean distances are shown for data points $\mathbf{y}_1$ and $\mathbf{y}_2$ in 2D space. The dashed line shows the Euclidean distance while the solid line represents the Manhattan distance.

- Mahalanobis distance: This measure considers correlations among the features as given by

$$d_{Mahalanobios}(\mathbf{y}_1, \mathbf{y}_2) = (\mathbf{y}_1 - \mathbf{y}_2)\Sigma^{-1}(\mathbf{y}_1 - \mathbf{y}_2)^T, \tag{2.24}$$

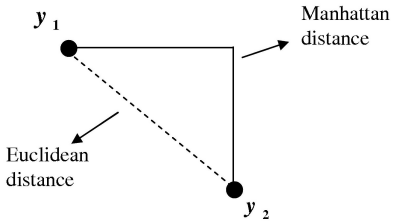


Figure 2.5: Euclidean and Manhattan distances

where Σ is the associated covariance matrix and T indicates the vector transpose.

- Cosine similarity: This measure indicates the cosine angle of two vectors defined as

$$d_{Cosine}(\mathbf{y}_1, \mathbf{y}_2) = \frac{\mathbf{y}_1 \cdot \mathbf{y}_2}{|\mathbf{y}_1||\mathbf{y}_2|} = \frac{\sum_{i=1}^{d} \mathbf{y}_{1,i} \cdot \mathbf{y}_{1,i}}{(\sum_{i=1}^{d} \mathbf{y}_{1,i}^2)^{1/2} \cdot (\sum_{i=1}^{d} \mathbf{y}_{2,i}^2)^{1/2}}. \tag{2.25}$$

  The more similar the points are, the greater the cosine value is.

Euclidean distance is considered as the distance function hereafter.

2.6 Cluster Validity Measures

These measures are usually used to evaluate the quality of clustering techniques [25]. In the following, we briefly explain some quality measures of clustering techniques.

2.6.1 Compactness measure

Compactness measure specifies that how much the samples of a cluster are similar to each other and are different from those in other clusters [2]. An appropriate example for this measure is *within-cluster distance* [10]. The compactness of clusters in terms of within-cluster distance is calculated by

$$\mathcal{F}_c(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{n_k} d(\mathbf{m}^{(k)}, \mathbf{y}_j^{(k)}), \tag{2.26}$$

where $d(\cdot)$ stands for the distance between cluster center, $\mathbf{m}^{(k)}$, and sample j of cluster k, $\mathbf{y}_j^{(k)}$. The goal is to minimize this measure as much as possible.

2.6.2 Separation measure

This criterion shows how far the clusters are from each other [10]. The clusters' separation measure can be defined by

$$\mathcal{F}_s(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) = \frac{1}{K(K-1)} \sum_{j=1}^{K} \sum_{k=j+1}^{K} d(\mathbf{m}^{(j)}, \mathbf{m}^{(k)}). \tag{2.27}$$

This measure, also known as *between-cluster* distance, computes the cumulative distance of cluster centers from each other. Clustering techniques aim to maximize this criterion or equivalently minimize $-F_s(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)})$.

2.6.3 Combined measure

This measure is a linear combination of the compactness and separation measures. Having the within-cluster and between-cluster distances defined, we can now construct the combined measure. Here, we deal with a multi-objective function containing two different functions namely $F_c(\cdot)$ and $F_s(\cdot)$. The former function should be minimized, whereas the later needs to be maximized. By knowing that $max\ f(\mathbf{x})$ is equivalent to $min\ (-f(\mathbf{x}))$, the weighted sum of the objective functions can be expressed as:

$$\mathcal{F}_{Combined}(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) = w_1 \mathcal{F}_c(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) - w_2 \mathcal{F}_s(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}), \tag{2.28}$$

where w_1 and w_2 are weighting parameters such that $w_1 + w_2 = 1$ [6], [18].

2.6.4 Turi's validity index

This index is defined as

$$\mathcal{F}_{Turi}(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) = (c \times \mathcal{N}(2,1) + 1) \times \frac{intra}{inter}, \tag{2.29}$$

where c is a user-specified parameter and $\mathcal{N}(\cdot)$ is a Gaussian distribution with mean two and standard deviation one. In this thesis, parameter c is set to one. The $intra$ denotes the within-cluster distance provided in equation (2.26). Furthermore, the $inter$ term is the minimum distance between the cluster centers given by

$$\begin{aligned} inter = \quad & min\{\|\mathbf{m}^{(k)} - \mathbf{m}^{(l)}\|\}, \\ & k \in [1, ..., K-1], \\ & l \in [k+1, ..., K]. \end{aligned} \tag{2.30}$$

The aim of the different clustering approaches is to minimize Turi's index [26].

2.6.5 Dunn's index

Let's define $\alpha(C^{(k)}, C^{(l)})$ and $\beta(C^{(k')})$ as

$$\begin{aligned} \alpha(C^{(k)}, C^{(l)}) &= \min_{\mathbf{x}\in C^{(k)}, \mathbf{z}\in C^{(l)}} d(\mathbf{x}, \mathbf{z}), \\ \beta(C^{(k')}) &= \max_{\mathbf{x}, \mathbf{z}\in C^{(k')}} d(\mathbf{x}, \mathbf{z}). \end{aligned} \tag{2.31}$$

Dunn's index [27] can now be computed as

$$\mathcal{F}_{Dunn}(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) = \min_{1\le k\le K}\{\min_{k+1\le l\le K}\{\frac{\alpha(C^{(k)}, C^{(l)})}{\max_{1\le k'\le K}\beta(C^{(k')})}\}\}. \tag{2.32}$$

Clustering techniques are required to maximize Dunn's index.

2.6.6 S_Dbw index

Let *Scatt* denotes the average scattering of the clusters as a measure of compactness expressed by

$$Scatt = K^{-1}\sum_{k=1}^{K}\frac{\|\sigma(C^{(k)})\|}{\|\sigma(Y)\|}, \tag{2.33}$$

where $\sigma(\cdot)$ stands for the variance of the associated data and $\|\mathbf{x}\|$ is defined as $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$. Then, the separation measure is defined as

$$Den\_bw = \frac{1}{K(K-1)}\sum_{k=1}^{K}\sum_{\substack{l=1\\ l\neq k}}^{K}\frac{\mathcal{D}(\mathbf{z}_{k,l})}{max\{D(\mathbf{m}^{(k)}), \mathcal{D}(\mathbf{m}^{(l)})\}}, \tag{2.34}$$

where $\mathbf{z}_{k,l}$ is the middle point of the line segment defined by cluster centers $\mathbf{m}^{(k)}$ and $\mathbf{m}^{(l)}$. Also, $\mathcal{D}(\mathbf{m}^{(k)})$ denotes a density function around point $\mathbf{m}^{(k)}$ which is estimated by $\mathcal{D}(\mathbf{m}^{(k)}) = \sum_{j=1}^{n_k} f(\mathbf{m}^{(k)}, \mathbf{y}_j^{(k)})$, and

$$f(\mathbf{m}^{(k)}, \mathbf{y}_j^{(k)}) = \begin{cases} 1 & \text{if } d(\mathbf{m}^{(k)}, \mathbf{y}_j^{(k)}) > \tilde{\sigma} \\ 0 & \text{Otherwise,} \end{cases} \tag{2.35}$$

where $\tilde{\sigma} = K^{-1}\sqrt{\sum_{k=1}^{K}\|\sigma(C^{(k)})\|}$. Finally, S_Dbw index [25], [28] is defined as

$$\mathcal{F}_{S\_Dbw}(\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)}) = Scatt + Den\_bw. \tag{2.36}$$

Maximizing this index is of interest when trying to cluster a set of data into several groups.

The combined measure is used as the required fitness function of the PSO procedure in the rest of the thesis. However, the other measures will be used to evaluate the proposed approach as well.

2.7 Summary

Extracting subgroups of a set of unlabeled data has become an important topic in machine learning and data mining. There exist various data clustering algorithms divided into hierarchical and partitional approaches, as studied earlier in this chapter. The first approach outputs a hierarchy of clusters whereas the later one generates a partition of the data.

Particle swarm clustering algorithms belong to the class of partitional approaches. Most of the traditional partitional approaches, such as K-means clustering, depend strongly on the initial conditions. This issue leads to convergence towards local optimal solutions. However, PSO-based clustering approaches probe the search space globally using population-based behavior. Consequently, they are more likely to escape from local optimal solutions. In the following chapter, our proposed approach for dealing with the data clustering problem is outlined.

Chapter 3

Multiple Cooperative Swarms Clustering

In this chapter, a novel clustering approach based on multiple particle swarms is presented. The multiple swarms clustering approach imitates the behavior of biological swarms which search for food situated in several places. This approach considers multiple cooperating swarms to find centers of clusters. This task is done in two phases: initialization and exploitation. In the initialization phase, the proposed approach assigns a portion of the search space to each swarm. In the exploitation phase, the search to reach a near-optimal solution proceeds using cooperating swarms. As compared to the single swarm clustering technique, multiple cooperative swarms provide better solutions in terms of fitness function measure for the centers of the clusters as the dimensionality of data and number of clusters increase. The performance of the proposed technique is also compared with that of existing clustering techniques for eight different data sets.

3.1 Introduction

Particle Swarm Optimization (PSO) is a search method that mimics the swarming behavior of flocks of birds [18], [19], and was first introduced by Kennedy and Eberhart [20], [29]. The same as Genetic Algorithms (GAs) [30], it employs a population of individuals known as particles to solve an optimization problem. As compared to GAs, a swarm is similar to a population, whereas a particle corresponds to an individual. PSO is used to optimize an objective function f, called fitness function. The PSO algorithm starts from an initial population and explores the search space through a number of iterations to reach a near-optimal solution.

PSO has been also used for solving data clustering problems [2], [3], [4], [8], [5], [23], [31], [32]. PSO-based clustering approaches probe the search space using a number of particles globally. However, most other clustering techniques perform a local search in which the solution obtained is situated in a narrow neighborhood of the previous solution [2].

The existing clustering methods based on particle swarm optimization consider only one swarm to explore the search space. However, there are several situations in which a single swarm is not able to search all of the space sufficiently, and so fails to return satisfactory results. In the cases where the dimensionality of data is high or there is a considerable number of clusters, multiple cooperative swarms can perform better than a single swarm, due to the exponential increase in the volume of the search space as the dimension of the problem increases.

The multiple cooperative swarms approach is a suitable tool to cluster data with high dimensions and large number of clusters. The core idea in the multiple cooperative swarms clustering approach is to use *divide and conquer strategy*. In other words, the whole search space is decomposed into several subdivisions each of which is coupled with a swarm, and then using cooperative approach among swarms the final solution is obtained.

3.2 Motivations

Here, the motivations behind applying particle swarm optimization for the clustering task are explained. We describe two main categories of motivations: biological and computational.

3.2.1 Biological motivations

Historically, the biological behavior of swarms was the main motivation behind particle swarm optimization [19], [33]. PSO founders mimicked the swarming behavior of flocks of birds. In the PSO algorithm, food (solution) is located in a single point and a swarm tends to reach that point. However, there are occasions in which there are several possible points to find food; for instance in the case of bees, usually there is more than one possible bunch of flowers. In other words, there is a cooperation between different species of bees to get nectar and different species of flowers to attract more bees [34].

Although PSO algorithm originated from the flocking behavior of birds, the swarming behavior of bees can be also considered in modeling the clustering task.

3.2.2 Computational motivations

Computational issues have also stimulated employing particle swarm optimization for clustering task. These motivations include:

- The PSO algorithm performs a global search of the solution space, whereas most other clustering techniques perform a local search [2]. In the local search, the solution obtained is located within the vicinity of the previous solution.

For example, the K-means clustering algorithm applies the randomly generated points as the initial centers of clusters and updates the position of the centers at every iteration. This may cause the algorithm to converge to sub-optimal solutions. At the same time, PSO is less sensitive to the effect of the initial conditions due to its population-based nature. Therefore, it is more probable to find near-optimal solutions.

- Particle swarm optimization has been used to solve multi-objective optimization problems [18], [35], [36], [37], [38], [39], [40], [41]. From an optimization point-of-view, clustering can be considered as a multi-objective problem. On the one hand, we desire to have as compact clusters as possible; on the other hand, we prefer to have as separate clusters as possible. Conventional clustering techniques such as K-means usually consider only the former criterion, whereas the PSO-based clustering technique can deal with multiple objectives [2], [3].

- Both multiple and cooperative swarms have also been introduced to resolve optimization problems [42], [43], [44]. Van den Bergh and Engelbrecht have used cooperative multiple swarms to solve optimization problems [42]. Their proposed method performs better than *canonical PSO* (or a single swarm-based technique versus multiple swarms) in high dimensions, due to the exponential increase in the volume of the search space as the dimension of the problem increases. This idea is valid for clustering problems as well. When the dimensionality of the data is high, the ability of a single swarm is not sufficient to search all of the solution space. Instead, multiple swarms cooperating together can be employed to obtain cluster centers effectively.

  There are two main approaches based on multiple particle swarms which are cooperative PSO and competitive PSO. In the former approach, some notion of cooperation is considered between different swarms. Cooperation is defined in terms of exchanging information about best solutions obtained so far by different swarms. In this approach, the success of one swarm enhances the overall performance of all swarms. In competitive PSO, however, there is the predator-prey relationship. A win for one swarm implies a failure for the other swarm. As a result, there is a direct competition in this approach [18].

  This thesis concentrates on the cooperative PSO. We distribute the search task among several swarms each of which traverses its associated region while cooperating with other swarms.

Having introduced the main motivations for the emerging approach of PSO-based clustering, we outline next the main characteristics of the proposed multiple cooperative swarms clustering approach. The latter approach,

3.3 Multiple Swarms Clustering

A new technique for the data clustering task using multiple swarms is provided. We start with the following assumptions:

- The number of swarms is equal to the number of clusters. That is, each swarm corresponds to a cluster.
- Each swarm is responsible for finding its related cluster's center.
- Particles of each swarm are candidates for the corresponding cluster's center.

The whole procedure to reach a near-optimal solution in the proposed approach is performed through two main phases: *initialization* and *exploitation*. Schematic representations of multiple swarms during the initialization and exploitation phases are depicted in Fig. 3.1 and Fig. 3.2, respectively. In this figure, a set of data points are given and the aim is to cluster these points into four distinct clusters. The swarm size is also set to five. In the initialization phase, there is a super-swarm which guides the other swarms. Also, there is no information exchange between swarms in this phase. Situation of swarms at the beginning and end of the initialization phase is given in parts (Fig. 3.1.a.1, Fig. 3.1.b.1) and (Fig. 3.1.a.2, Fig. 3.1.b.2), respectively. At the beginning of the exploitation phase, the cooperation between multiple swarms initiates and each swarm investigates its associated region (Fig. 3.2.a.3 and Fig. 3.2.b.3). When the particles of each swarm converge as observed in (Fig. 3.2.a.4), the final solution for cluster centers is released (Fig. 3.2.b.4).

A comprehensive explanation of the initialization and exploitation phases is given next.

1. Initialization phase

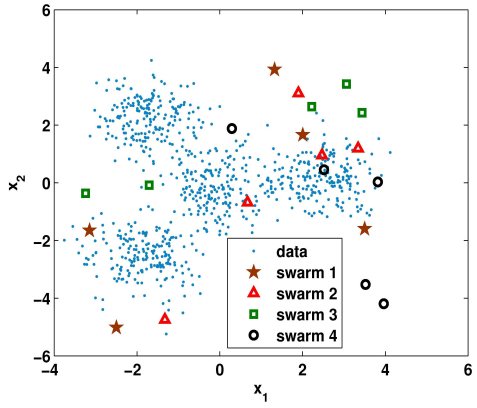
   The search starts from random points in the solution space. At the beginning of this phase, there are overlaps between swarms, whereas at the end each swarm will deal with a part of the search space. The situation of swarms at the beginning and end of the initialization phase is shown in Fig. 3.1.a.1 and Fig. 3.1.a.2, respectively.

   A part of the search space explored by a swarm is called a *swarm region*. Each swarm region is characterized by two parameters:

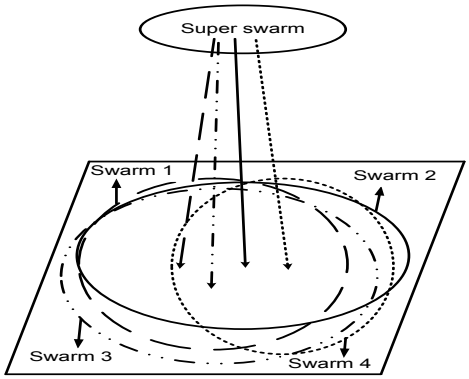
   - Center of swarm region, $\mathbf{z}^{(k)},\ k \in [1, \cdots, K]\cdot$
   - Width of swarm region, $R^{(k)},\ k \in [1, \cdots, K]\cdot$

   The first parameter shows the center of the swarm region and the second one gives its corresponding radius. The main goal of the initialization phase is to determine these parameters for all swarms.

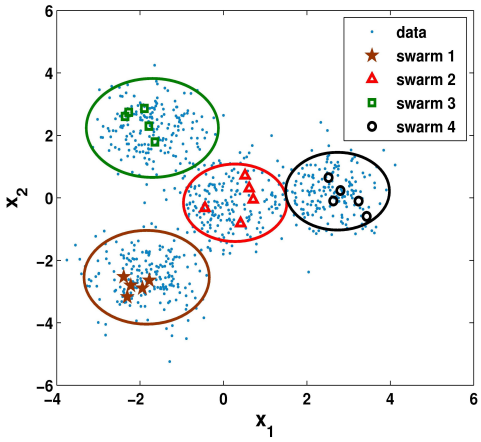
a.1. Before initialization:
swarms in all search space



b.1. Before initialization:
super-swarm and other swarms



a.2. After initialization:
swarms' region is determined



b.2. After initialization:
super-swarm and other swarms

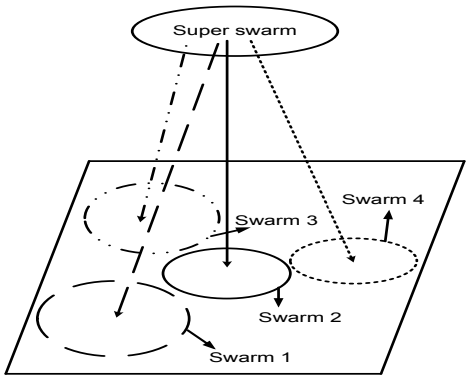
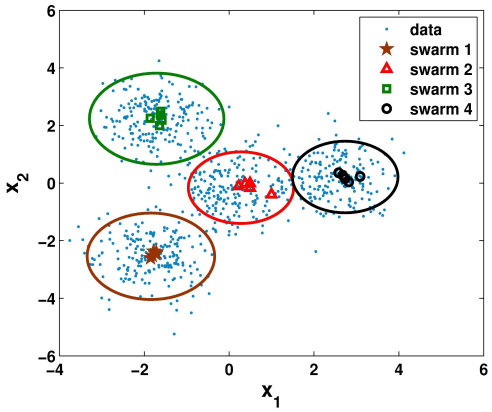
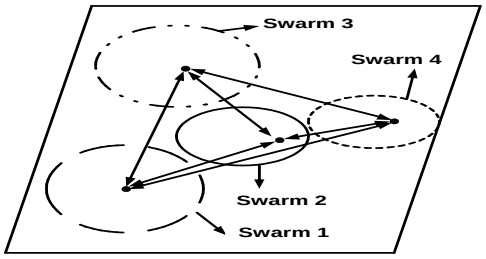


Figure 3.1: Schematic representation of multiple swarms during the initialization phase

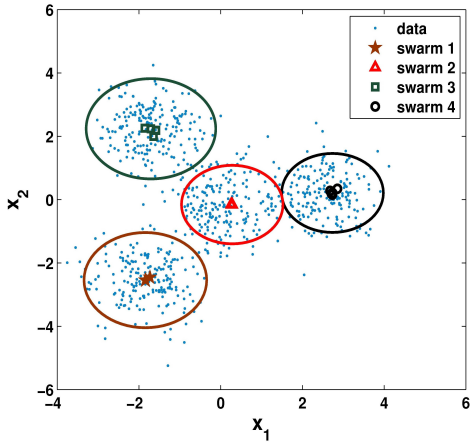
a.3. Beginning of exploitation: explore the search space



b.3. Beginning of exploitation: cooperation between swarms



a.4. End of exploitation: convergence



b.4. End of exploitation: final solution

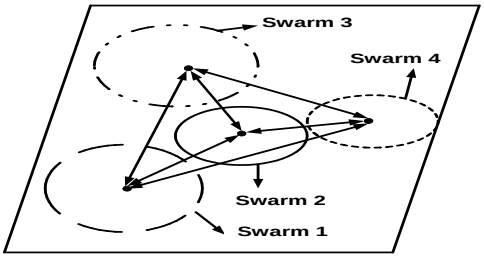


Figure 3.2: Schematic representation of multiple swarms during the exploitation phase

To perform the initialization phase, we consider another swarm, called a *super-swarm*. The super-swarm obeys the single swarm clustering technique to direct swarms to dense locations. The relation of the super-swarm to the other swarms before and after initialization phase is illustrated in Fig. 3.1.b.1 and Fig. 3.1.b.2, respectively.

In the initialization phase, each swarm receives information only from the super-swarm. Let $\Phi_i = (\varphi^{(1)}, ..., \varphi^{(K)})_i$ denotes particle i of super-swarm, where $\varphi^{(k)}$ is the candidate for center of swarm region k. Moreover, let us assume $\Phi^* = (\varphi^{(1)}, ..., \varphi^{(K)})^*(t)$ denotes the global best of the super-swarm at time step t, and let denote its k^{th} element by $(\varphi^{(k)})^*$.

First, the super-swarm searches for the center of the swarm regions $\Phi_i = (\varphi^{(1)}, ..., \varphi^{(K)})_i$. After updating the positions and velocities, the global best of the super-swarm is determined. The updated global best information is then supplied to all swarms. In other words, the global best of swarm k is defined as

$$(\mathbf{x}^{(k)})^* = (\varphi^{(k)})^*. \tag{3.1}$$

Thus, each swarm k tries to move toward $(\varphi^{(k)})^*$. Also, one of the cluster validity measures described in chapter 2 is considered as the fitness function.

After all swarms have updated the position and velocity of their particles, a new iteration commences. Again, the super-swarm updates the centers of the swarm regions. These new centers are fed to the swarms. The initialization phase ends when the centers and widths of the swarm regions do not change over successive iterations, or the maximum number of iterations is achieved.

At the end of the initialization phase, the center of the swarm region is determined by

$$\mathbf{z}^{(k)} = (\varphi^{(k)})^* \ , \quad k \in [1, ..., K]. \tag{3.2}$$

Furthermore, the width of the swarm region needs to be computed for all swarms at the end of this phase. To determine the width, we propose using an eigen decomposition theorem. Let's assume $\lambda_{max}^{(k)}$ denotes the square root of the biggest eigen value of data points belonging to swarm k, which is computed by using the center of the swarm regions as initial cluster centers. The width of the swarm region k is then computed by:

$$R^{(k)} = \alpha \lambda_{max}^{(k)} \ , \quad k \in [1, ..., K], \tag{3.3}$$

where α is a positive constant. The α is selected such that an appropriate coverage of the search space, or feasible solution region, is obtained. The scheme c of Fig. 3.3 provides such coverage as there is no overlap between different swarm regions and maximum coverage of the search space is attained.

For example, changes of the fitness function in terms of Turi's validity index with α for the wine data set explained in section 5 of this chapter is shown in Fig. 3.4. According to this figure, the best value for α is observed at 0.25.

Algorithm 3.1 Pseudocode for the initialization phase of multiple swarms clustering

initialize super-swarm of size n
initialize K swarms of size n
repeat
  for each particle $i \in [1, \cdots, n]$ of super-swarm **do**
    update position and velocity
    if $F(\Phi_i(t+1)) < F(\Phi_i^{pb}(t))$ **then**
      $\Phi_i^{pb}(t) \leftarrow \Phi_i(t+1)$
    end if
  end for
  $\Phi^*(t) \leftarrow argmin\{F(\Phi_i^{pb}(t))|\ i \in [1, \cdots, n]\}$
  for each swarm $k \in [1, \cdots, K]$ **do**
    $(\mathbf{x}^{(k)})^*(t) \leftarrow (\varphi^{(k)})^*(t)$
    for each particle $i \in [1, \cdots, n]$ of swarm k **do**
      update position and velocity
    end for
  end for
until termination criterion is met
for each swarm $k \in [1, \cdots, K]$ **do**
  $\mathbf{z}^{(k)} \leftarrow (\varphi^{(k)})^*(T)$
  $R^{(k)} \leftarrow \alpha\lambda_{max}^{(k)}$
end for

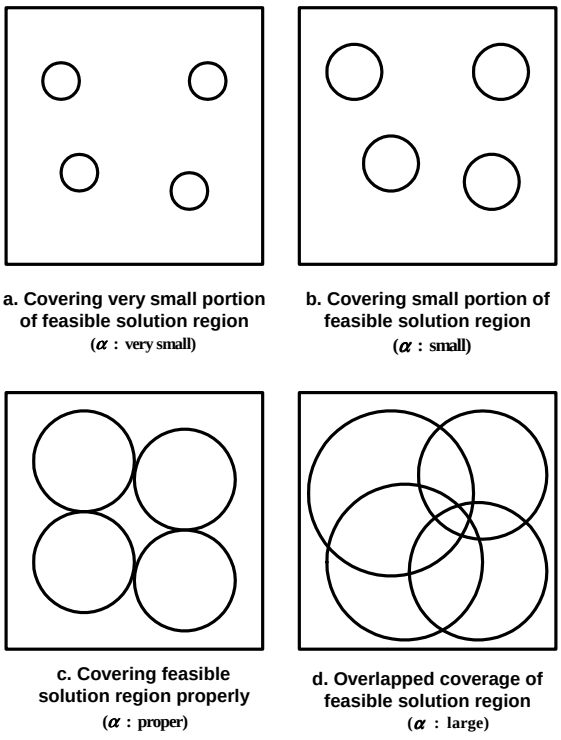


Figure 3.3: Different schemes for width of swarm regions

The required pseudocode for the initialization phase is presented in Algorithm **3.1**.

2. Exploitation phase

   After initializing the swarms, each swarm explores for the best solution as cluster center within its corresponding region. In this phase, there exists no super-swarm, but rather there is information exchange among swarms. Hence, there is cooperation among swarms to find the final solution. Each swarm knows the global best of the other swarms.

   This phase contains a number of iterations converging on a near-optimal solution. Each iteration is composed of two main steps: *search*, and *make decision*. In the search step, the search within each swarm region proceeds. In the make decision step, it is revealed whether the new solution is acceptable or not. Description of these steps are provided here.

   (a) Search

   During this process, the search within each swarm region is done such that the within-cluster distance is minimized, while at the same time the accumulated distance from other clusters is maximized. The *compactness of cluster k* given

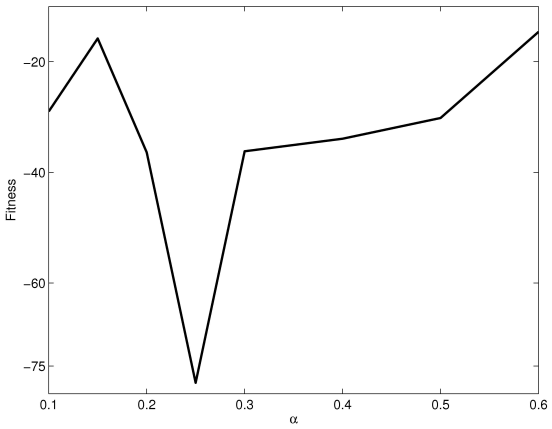


Figure 3.4: Finding proper α for wine data (introduced in section 3.5)

particle i as its center is defined as

$$f_c(\mathbf{x}_i^{(k)}) = \frac{1}{n_k} \sum_{j=1}^{n_k} d(\mathbf{x}_i^{(k)}, \mathbf{y}_j^{(k)}), \tag{3.4}$$

where $\mathbf{x}_i^{(k)}$ is particle i of swarm k. Also, $d(\cdot)$ stands for the distance between the cluster center, $\mathbf{x}_i^{(k)}$, and the cluster's data point, $\mathbf{y}_j^{(k)}$.

Distance from other clusters shows how far that particular cluster is from other clusters. This distance for particle i of swarm k can be formulated as follows:

$$f_s(\mathbf{x}_i^{(k)}) = \frac{1}{K-1} \sum_{j=1, j\neq k}^{K} dist(\mathbf{x}_i^{(k)}, \mathbf{m}^{(j)}) \cdot \tag{3.5}$$

Thus, the objective function for particle i of swarm k, $\mathbf{x}_i^k$, is given by

$$f(\mathbf{x}_i^{(k)}) = w_1 f_c(\mathbf{x}_i^{(k)}) - w_2 f_s(\mathbf{x}_i^{(k)}), \tag{3.6}$$

where w_1 and w_2 are weighting parameters such that $w_1 + w_2 = 1$. After defining the objective function, the mathematical model of the clustering -in terms of the optimization problem using multiple swarms- can be constructed. In search step within each swarm k, particles attempt to minimize the following optimization problem:

$$\begin{gathered} \min \ f(\mathbf{x}_i^{(k)}) = w_1 f_c(\mathbf{x}_i^{(k)}) - w_2 f_s(\mathbf{x}_i^{(k)}), \\ s.t. : \|\mathbf{x}_i^{(k)} - \mathbf{z}^{(k)}\| \leq R^{(k)}, \\ i \in [1, \cdots, n], \end{gathered} \tag{3.7}$$

where $\|.\|$ stands for Euclidean norm. In this equation, the constraint forces particles of the swarm to search within the corresponding swarm region. A new position for a particle is accepted only if it is inside the swarm region.

The search using multiple swarms is performed serially. It begins in the first swarm region, where a new candidate for the cluster center ($\mathbf{m}'^{(1)}$) is obtained using equation (3.7). Considering this new candidate, the next swarm searches for a new candidate for its corresponding cluster center ($\mathbf{m}'^{(2)}$). Similarly, this procedure is repeated for each of the following swarms to obtain new candidates for the centers of all clusters, $\mathbf{M}' = (\mathbf{m}'^{(1)}, \cdots, \mathbf{m}'^{(K)})$.

(b) Make decision

When the search for all swarms is completed, it is necessary to decide on the new candidates for the cluster centers. In other words, a final decision is made on accepting or rejecting the solution proposed by multiple swarms. If the fitness value obtained by equation (2.28) for new candidates $\mathbf{M}' = (\mathbf{m}'^{(1)}, \cdots, \mathbf{m}'^{(K)})$ for cluster centers is smaller than the fitness value of the former centers $\mathbf{M} = (\mathbf{m}^{(1)}, \cdots, \mathbf{m}^{(K)})$, the new solution is valid and accepted; otherwise, it is rejected.

The required pseudocode for the exploitation phase is given in Algorithm **3.2**.

Algorithm 3.2 Pseudocode for the exploitation phase of multiple swarms clustering

```
initialize K swarms of size n such that the positions and velocities of each swarm
are selected inside the associated swarm's region.
repeat
  for each swarm k ∈ [1, ··· , K] do
    for each particle i ∈ [1, ··· , n] of swarm k do
      update position and velocity
      if f(x_i^(k)(t+1)) < f(x_i^(k),bp(t)) then
        x_i^(k),bp(t) ← x_i^(k)(t+1)
      end if
    end for
    m'^(k)(t) ← arg min{f(x_i^(k),bp(t)) | i ∈ [1, ··· , n]}
  end for
  if F(M'(t)) < F(M(t)) then
    M(t) ← M'(t)
  end if
until termination criterion is met
```

Having outlined the procedure in full, we now provide the overall algorithm of the proposed method in Algorithm **3.3**.

In the following, the contribution of the initialization phase and cooperation among swarms in the proposed approach are demonstrated.

Algorithm 3.3 Multiple swarms clustering

Phase 1: Initialization by the super-swarm

- Determine swarms' center, $\mathbf{z}^{(k)},\ k \in [1, \cdots, K]\cdot$
- Determine swarms' width, $R^{(k)},\ k \in [1, \cdots, K]\cdot$

Phase 2: exploitation

- Step 1: Search within each swarm
  - 1.1. Compute new positions of all particles of swarms.
  - 1.2. Obtain the fitness value of all particles using equation (3.6).
  - 1.3. Select the position which minimizes the optimization problem using equation (3.7) and denote it as the new candidate for corresponding cluster center ($\mathbf{m}'^{(k)}$).
- Step 2: Make decision
  - 2.1. Calculate the fitness value of the new candidates for centers of clusters ($\mathbf{m}'^{(1)}, \cdots, \mathbf{m}'^{(K)}$) using equation (2.28).
  - 2.2. If the fitness value is smaller than that of previous iteration, accept the new solution; otherwise, reject it.
  - 2.3. If termination criterion is achieved, stop; otherwise, go back to step 1.

3.3.1 Proposed approach without initialization

In order to examine the impact of the initialization phase, the proposed approach is considered without an initialization phase. Assume that the data points provided in Fig. 3.1 are going to be clustered into four clusters, and that there exist four different swarms. By applying the proposed approach without an initialization phase, the situation of particles of all swarms at the beginning, middle and end of the exploitation phase is displayed in Fig. 3.5.

By eliminating the initialization phase, the distribution of the search space among swarms is meaningless, and consequently each swarm deals with the whole of search space. The performance of the proposed approach with and without initialization is presented in Fig. 3.6 in terms of the compactness, separation and combined measures. The solid and dashed lines in Fig. 3.6.a and Fig. 3.6.b are obtained by testing the proposed approach with and without initialization phase respectively for 30 independent runs. Moreover, μ in Fig. 3.6.a and Fig. 3.6.b denotes the average value of the 30 runs for each measures. By comparing the obtained average values, it is clear that the presence of the initialization phase enables the proposed approach to provide better results in terms of compactness and separation measures. Fig. 3.6.c shows the average values of the combined measure for 30 independent runs over 80 iterations with and without initialization phase. It verifies that the proposed approach with initialization phase returns better solutions in terms of the combined measure.

Our experiments using the T-test [45] indicates that the difference between the proposed approach and the approach without an initialization phase in terms of the compactness and combined measures is statistically significant at a significance level 5%.

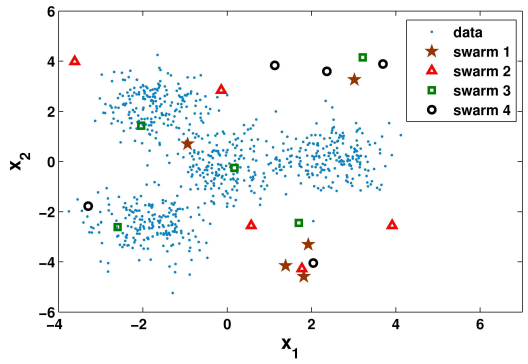
3.3.2 Proposed approach without cooperation

Now, let us evaluate the effect of cooperation on the proposed approach. Assume there is no cooperation among the swarms during the exploitation phase. By repeating the procedure of the proposed approach without cooperation using the data points provided in Fig. 3.1, the position of particles of the swarms during the initialization and exploitation phases is illustrated in Fig. 3.7.

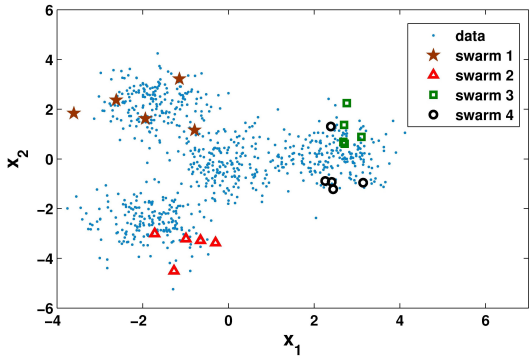
By disregarding cooperation among swarms, there is no information exchange between swarms and each swarm updates its corresponding cluster's center without knowing other swarms outputs at each iteration. That is, particles of each swarm k do not know the center of other swarms. Consequently, they obtain new positions by ignoring f_s component in equation (3.7).

The influence of excluding cooperation from the proposed approach is investigated in Fig. 3.8 according to the compactness, separation and combined measures.

a. Beginning



b. Middle



c. End

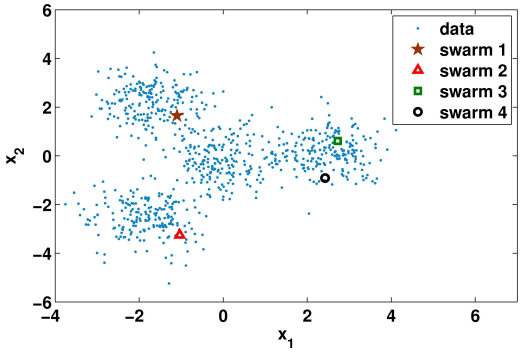
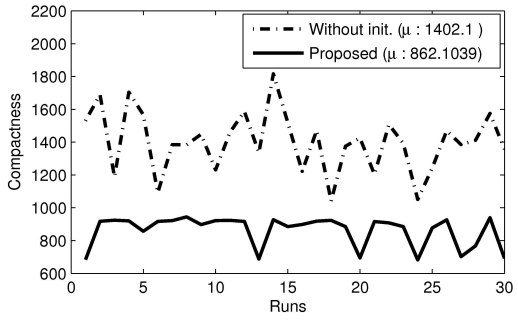
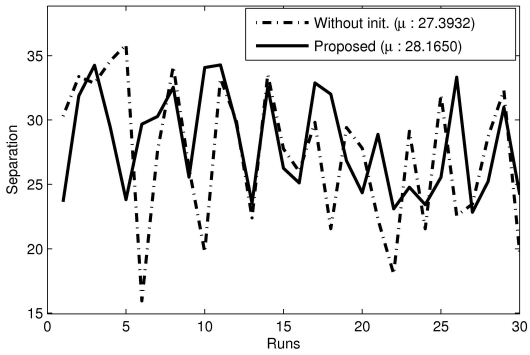


Figure 3.5: The situation of particles of all swarms at different stages of the proposed approach without initialization phase

a. Compactness



b. Separation



c. Combined measure

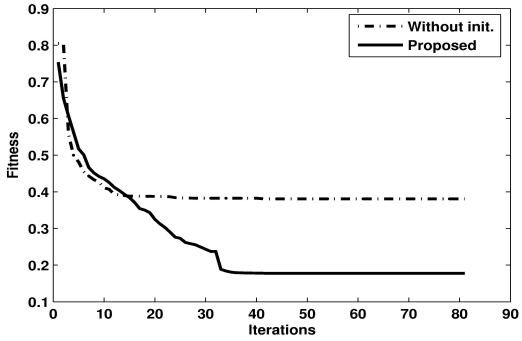
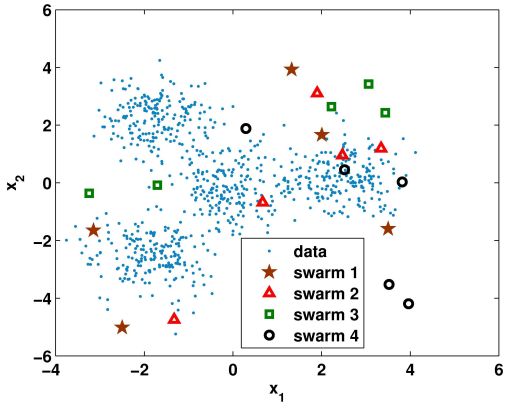
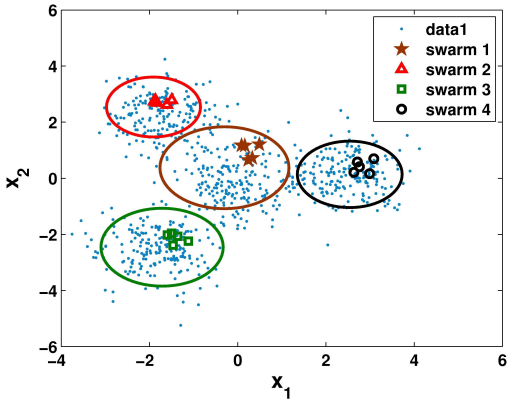


Figure 3.6: Evaluating the influence of removing initialization phase according to the compactness, separation and combined measures

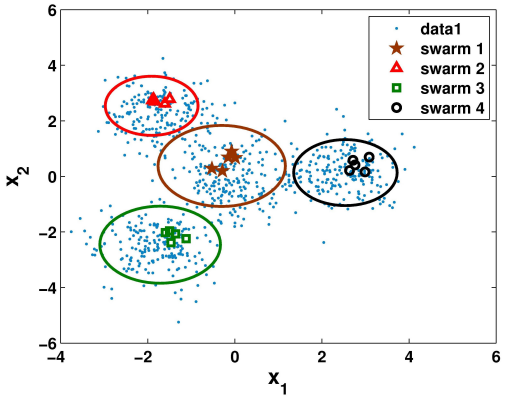
a. Before initialization:
swarms in all search space



b. After initialization:
swarms' region is determined



c. Beginning of exploitation:
explore the search space



End of exploitation:
no convergence

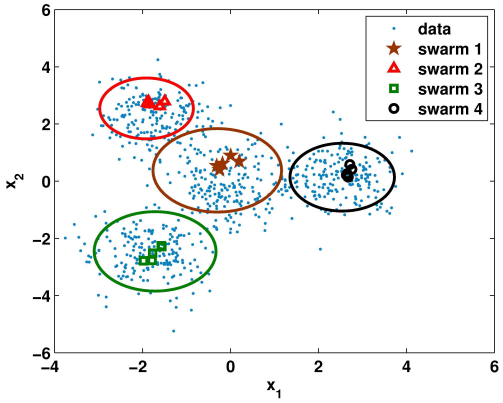
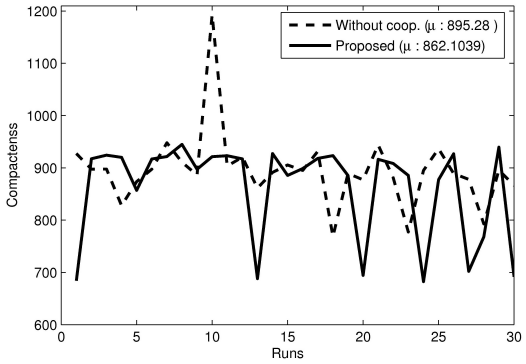
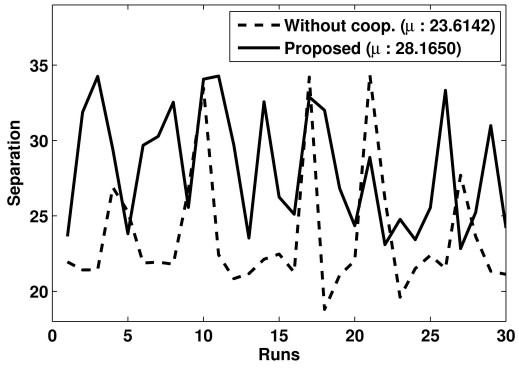


Figure 3.7: The situation of particles of all swarms at different stages of the proposed approach without cooperation in the exploitation phase

a. Compactness



b. Separation



c. Combined measure

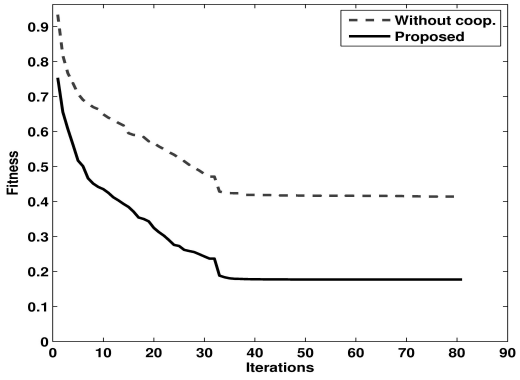


Figure 3.8: Evaluating the influence of removing cooperation according to the compactness, separation and combined measures

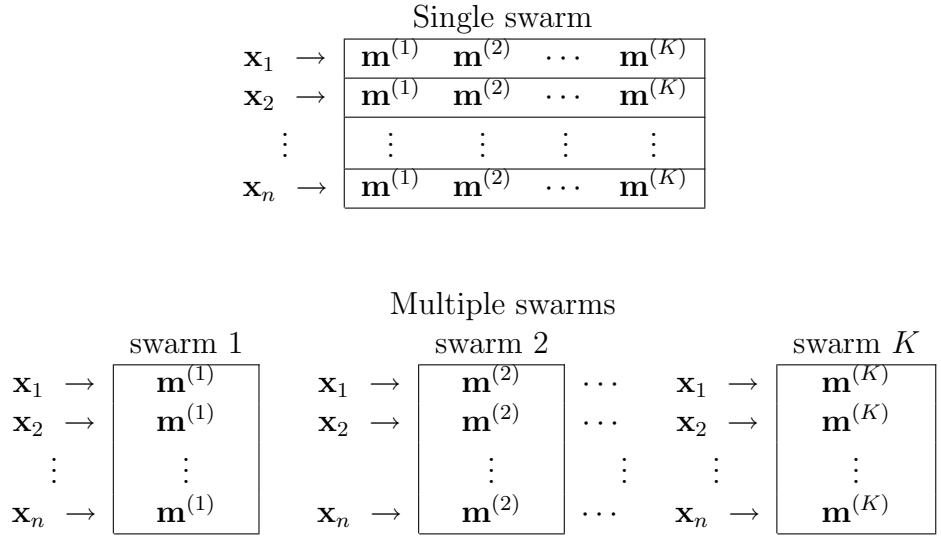


Figure 3.9: Comparing the computational complexity of single swarm and multiple swarms approaches with regard to particle definition

Also, experiments using the T-test revealed that the difference between the proposed approach and the approach without cooperation in terms of the separation and combined measures is statistically significant at a significance level 5%.

3.3.3 Computational complexity

At first glance, it may be thought that the computational complexity of the proposed approach exceeds that of the single swarm clustering, as it employs multiple swarms rather than a single swarm. However, a more precise inspection acknowledges that these two approaches are computationally more or less the same in terms of the number of particles' components. Let's recall the definition of particles in both approaches. Assume n denotes the size of the swarm for both cases. A particle $\mathbf{x}_i$ in single swarm approach is defined by $\mathbf{x}_i = (m^{(1)}, \cdots, m^{(K)})_i,\ i \in [1, \cdots, n]$. In other words, each particle has K different components. Therefore, the total number of components in a single swarm is $n \cdot K$.

In the multiple swarms approach, a particle is defined by $x_i^{(k)} = m_i^{(k)},\ i \in [1, \cdots, n],\ k \in [1, \cdots, K]$. That is, each particle possesses only one component. Since each swarm includes n particles and there exist K different swarms, the total number of components will be $n \cdot K$, the same as the single swarm approach (Fig 3.9).

Furthermore, there is no super-swarm in the exploitation phase as it is only used temporarily for initializing swarms. In terms of convergence rate, having cooperation between swarms accelerates the speed of arriving at the final solution.

There is also another difference between these approaches from a computational point-of-view. In the single swarm approach, the swarm explores the whole search

space to provide a candidate for each component or cluster center. In contrast, the proposed approach distributes the search task among several swarms such that each swarm is responsible to probe a part of the search space to yield its solution for the cluster center. For high dimensional data and in those cases where there is a large number of clusters, the probability of getting an optimal solution using a single swarm decreases. That is because the volume of the search space exponentially increases as the dimension of the data increases.

The computational complexity of different approaches in terms of run time has been also studied for the data set shown in Fig. 3.1. The proposed and single swarm-based approaches provide the clustering result at about 1.4 seconds. This time for hybrid PSO, K-means, k-harmonic means and fuzzy c-means is about 1.2, 0.35, 94, and 4.9 seconds, respectively, for a single run.

3.4 Assessment of Multiple Cooperative Swarms Clustering

In this section, the performance of the proposed approach is evaluated and compared with other approaches such as K-means, K-harmonic means, fuzzy c-means, hybrid PSO and single swarm clustering approaches.

To examine the performance of the proposed approach, the following data sets have been used:

- Gaussian data: a total of 800 samples drawn from four two-dimensional Gaussian classes [8] with the following distributions:

$$N(\mu = \begin{bmatrix} m_i \\ 0 \end{bmatrix}, \sum = \begin{bmatrix} 0.50 & 0.05 \\ 0.05 & 0.50 \end{bmatrix}), \tag{3.8}$$

  where μ denotes the mean vector and $\sum$ is the covariance matrix, $m_1 = -3,\ m_2 = 0,\ m_3 = 3$ and $m_4 = 6$.

- seven data sets from UCI machine learning repository [46]: In Table 3.1, these data sets and the associated information of each data set such as the number of classes, number of samples and dimensionality are provided.

To illustrate the concept of initialization and exploitation, the proposed approach has been applied to Gaussian data over 130 iterations. The results provided in Fig. 3.10 indicate the mean and standard deviation of the fitness values for 30 independent runs. The mean (mu) of the fitness value is shown by solid line and the associated standard deviation (σ) is also represented by dash and dot lines.

As shown in Fig. 3.10, the initialization phase is terminated after 30 iterations. Due to the presence of cooperation between multiple swarms, a significant improvement is observed at the beginning of the exploitation phase.

Table 3.1: Data sets chosen from UCI repository

Data set	classes	samples	dimensionality
Iris	3	150	4
Wine	3	178	13
Teaching assistant evaluation	3	151	5
Breast cancer	2	569	30
Zoo	7	101	17
Glass identification	7	214	9
Diabetes	2	768	8

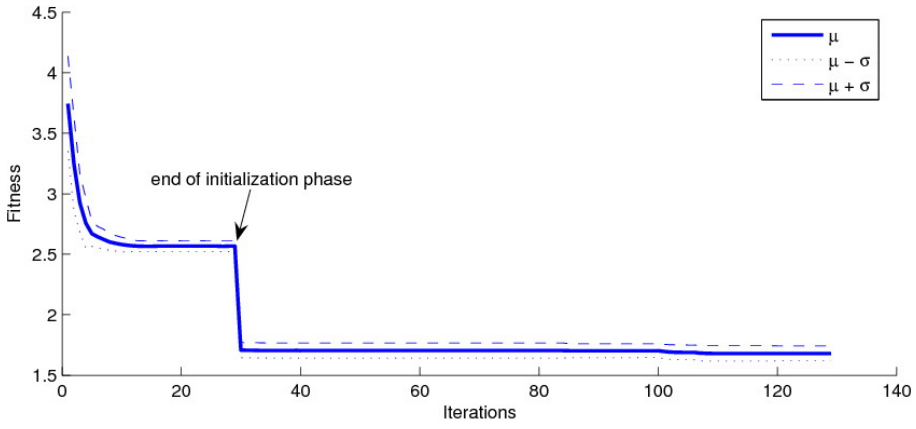


Figure 3.10: Convergence of the proposed approach in terms of combined measure as a fitness function

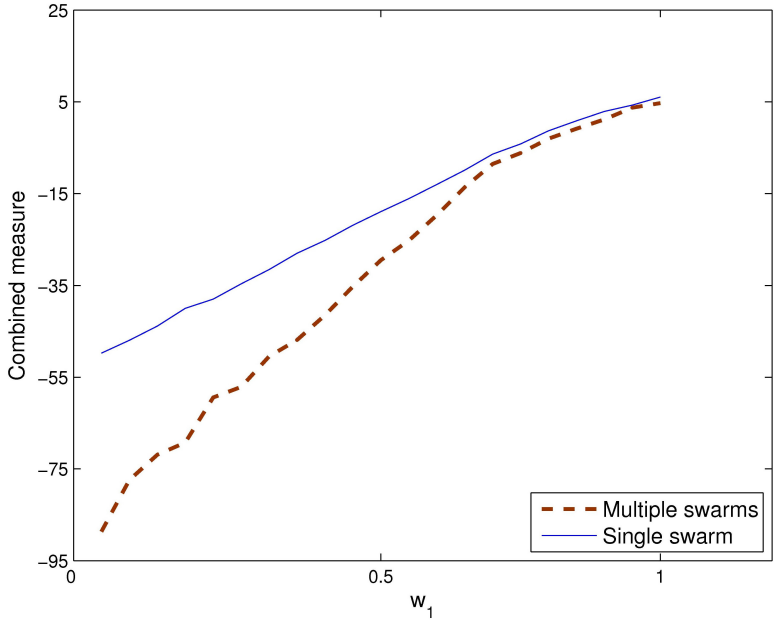


Figure 3.11: Sensitivity analysis for combined measure as a function of w_1

The parameters in the model are considered as $w = 1.2$ (decreasing gradually [22]), $c_1 = 1.49$, $c_2 = 1.49$ [22], and $n = 10$ (for all swarms). Moreover, the sensitivity of the proposed and single swarm approaches for w_1 appeared in the combined measure are provided in Fig. 3.11 using Gaussian data. As shown in this figure, the proposed approach is able to provide better solutions in terms of the combined measure for different values of w_1. As a result, the value of this parameter can be selected with regard to user preferences. It is set to be close to unity for the problems where the compactness has higher weight, whereas it is fixed to near zero values for the problems that separation is more important. In our experiment we have considered $w_1 = 0.85$ to keep a balance between compactness and separation measures. This rate can also be considered as a default value for w_1 in situations where a limited information about the underlying data is available.

Moreover, the number of clusters is considered to be equal to the number of classes for all data sets. In addition, the values of parameter α, defined in equation 3.3, are presented in Table 3.2 for all data sets and different measures.

In this table, *Comp.*, *Sep.*, *Comb.* and *Turi* terms indicate *compactness*, *separation*, *combined measure* and Turi's validity index, respectively.

3.4.1 Comparing the proposed approach with others

The proposed approach is compared with K-means (KH), K-harmonic means (KHM), fuzzy c-means(FCM), hybrid PSO and single swarm clustering using different data

Table 3.2: The values of parameter α for all data sets and different measures

Data set	Comp.	Sep.	Comb.	Turi
Gaussian	1	1.8	1.4	1.4
Iris	1.5	3	2	3
Wine	0.1	0.5	0.3	0.25
Teaching assistant evaluation	3	2	0.5	1.9
Breast cancer	0.1	0.4	0.35	0.2
Zoo	2	0.5	2	0.3
Glass identification	0.6	5.5	2	1.9
Diabetes	1.5	5.5	1.1	2.5

Table 3.3: Average and standard deviation comparison of diffirent measures for Gaussian data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	0.8409[0.1176]	-4.9533[0.0601]	-0.01[0.08]	0.2527[0.558]
KHM	0.816[0.001]	-3.0133[0.001]	0.39[0.168]	0.99e05[1.3e05]
FCM	0.8133[0.001]	-2.9842[0.001]	0.33[0.08]	0.2629[0.2879]
Hybrid PSO	0.8135[0.01]	-9.62[0.15]	-0.026[0.13]	0.3557[0.2723]
Single swarm	0.8044[0.0134]	$\mathbf{-9.9362}$[0.1988]	-0.03[0.1]	-63.44[5.008]
Multiple swarms	**0.8029**[0.0138]	-9.8527[0.1235]	$\mathbf{-0.04}$[0.02]	$\mathbf{-72.75}$[12.82]

sets. The results are provided in Tables 3.3-3.10. The comparisons are based on four validity measures defined in chapter 2, namely compactness, separation, combined and Turi's ones. The results have been obtained by averaging over 30 independent runs and the associated standard deviation ([σ]) for each value has also been provided.

As presented in Tables 3.3-3.10, the following observations can be declared:

- The multiple swarms clustering approach provides smaller values for both *combined measure* and *Turi's validity index* as compared to the other approaches for most of the data sets. Hence, the proposed approach is suitable in cases dealing with multiple objectives.
- In terms of the *separation* measure, both multiple swarms and single swarm clustering outperform other clustering approaches. In other words, these approaches can be used where the separate clusters are desired.
- In terms of the *compactness* measure, the proposed technique provides better results for all data sets as compared to the single swarm technique, though it is inferior to other clustering approaches for some of the data sets.

In Fig. 3.12 and Fig. 3.13, the fitness values in terms of *combined measure* for the proposed approach are compared with those of K-means , K-harmonic means, fuzzy

Table 3.4: Average and standard deviation comparison of diffirent measures for iris data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	0.6441[0.0739]	−3.2481[0.1917]	0.0881[0.0907]	0.3187[0.4853]
KHM	0.9312[0.6563]	−3.0199[1.2292]	0.3668[0.7358]	$0.8e05$[$1.1e05$]
FCM	**0.6071**[0.068]	−3.1682[0.0964]	0.0678[0.196]	0.43[0.39]
Hybrid PSO	0.6212[0.051]	−3.3853[0.3066]	0.0637[0.0447]	0.278[0.38]
Single swarm	0.6618[0.0528]	−5.9405[0.1104]	0.0436[0.0575]	−0.88[0.4]
Multiple swarms	0.6123[0.013]	**−6.0017**[0.135]	**0.023**[0.015]	**−0.89**[1.01]

Table 3.5: Average and standard deviation comparison of diffirent measures for wine data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	88.089[2.908]	−492.9[13.6854]	6.64[2.0059]	0.3623[0.3371]
KHM	142.3[47.06]	−514.6[176.31]	23.20[1.9]	$1.0e05$[$1.0e05$]
FCM	88.63[0.06]	−508.21[0.22]	3.37[0.02]	0.3561[0.3394]
Hybrid PSO	87.16[0.01]	−500.94[16.7]	9.16[0.07]	0.259[0.337]
Single swarm	86.52[0.38]	−962.37[13.3]	−9.4277[3.38]	−0.38[0.4]
Multiple swarms	**86.39**[0.229]	**−965.16**[3.66]	**−11.416**[0.86]	**−0.78**[0.8]

Table 3.6: Average and standard deviation comparison of diffirent measures for teaching assitant evaluation data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	9.85[0.228]	−20.9265[1.5992]	5.4544[0.4157]	0.68[0.75]
KHM	9.899[0.01]	−22.9464[0.01]	5.2066[0.01]	$1.12e06$[$1e06$]
FCM	10.125[0.2]	−15.3287[0.1731]	6.4891[0.04]	1.018[0.939]
Hybrid PSO	9.73[0.11]	−21.57[1.3489]	5.22[0.349]	0.63[0.61]
Single swarm	9.73[0.145]	**−54.9**[0.66]	4.23[0.1348]	**−0.56**[0.65]
Multiple swarms	**9.61**[0.05]	−54.6624[0.71]	**4.2041**[0.08]	−0.76[0.72]

Table 3.7: Average and standard deviation comparison of diffirent measures for breast cancer data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	252.3[22.6]	$-1.31e03$[20.16]	252.27[11.03]	0.23[0.23]
KHM	475.4[207.1412]	$-0.69e03$[0.8685]	234.04[17.92]	1.25[0.01]
FCM	**249.3**[0.0713]	$-1.297e03$[0.002]	28.447[0.22]	0.19[0.18]
Hybrid PSO	252.3[0.01]	$-1.315e03$[0.01]	−210.1[0.01]	0.17[0.25]
Single swarm	254.7[7.0324]	$\mathbf{-4.836}e03$[30.38]	−247.71[6.55]	−0.62[0.79]
Multiple swarms	252.0[0.01]	$-4.76e03$[67.8615]	**−248.8**[6.49]	**−0.66**[0.65]

Table 3.8: Average and standard deviation comparison of diffirent measures for zoo data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	1.1586[0.0592]	-4.5284[0.2805]	-0.10[0.13]	0.6208[1.0186]
KHM	1.1699[0.0841]	-4.8446[0.3042]	-0.139[0.1027]	0.4296[0.4803]
FCM	1.2684[0.0702]	-3.6786[0.3145]	0.1116[0.1025]	9.59[24.4366]
Hybrid PSO	1.1705[0.0893]	-5.7395[0.4116]	-0.126[0.1158]	0.9184[0.7134]
Single swarm	1.7277[0.0659]	$-$**8.87**[0.21]	-0.11[0.103]	-5.6769[3.7316]
Multiple swarms	**1.1582**[0.027]	-8.7389[0.1964]	$-$**0.17**[0.16]	$-$**6.1268**[4.41]

Table 3.9: Average and standard deviation comparison of diffirent measures for glass identification data

Method	Compactness[σ]	Separation[σ]	Combined [σ]	Turi's index[σ]
KM	**0.9276**[0.0373]	-4.23[0.52]	0.191[0.09]	1.32[1.08]
KHM	0.97[0.008]	-5.24[0.03]	0.08[0.0117]	$0.89e05$[$1.15e05$]
FCM	1.02[0.013]	-2.72[0.106]	0.4892[0.017]	6.68[0.0149]
Hybrid PSO	1.17[0.12]	-7.85[0.68]	$-$**0.016**[0.1454]	0.452[0.38]
Single swarm	1.47[0.11]	-11.119[0.30]	0.12[0.12]	-4.455[1.66]
Multiple swarms	1.21[0.14]	$-$**11.26**[1.26]	0.0227[0.0324]	$-$**5.205**[2.35]

Table 3.10: Average and standard deviation comparison of diffirent measures for diabetes data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KM	64.89[0.01]	-220.62[0.01]	24.1016[0.01]	0.28[0.26]
KHM	88.13[10.18]	-255.47[173.1]	87.053[3.16]	$1.6e07$[$2.6e07$]
FCM	**60.87**[0.0004]	-181.95[0.006]	26.18[0.0005]	0.337[0.33]
Hybrid PSO	64.89[0.01]	-220.62[0.01]	24.1[0.01]	0.27[0.32]
Single swarm	64.94[5.08]	-865.83[5.748]	-44.15[2.208]	-0.22[0.26]
Multiple swarms	61.41[0.88]	$-$**883.05**[115.8]	$-$**45.36**[1.27]	$-$**0.3**[0.31]

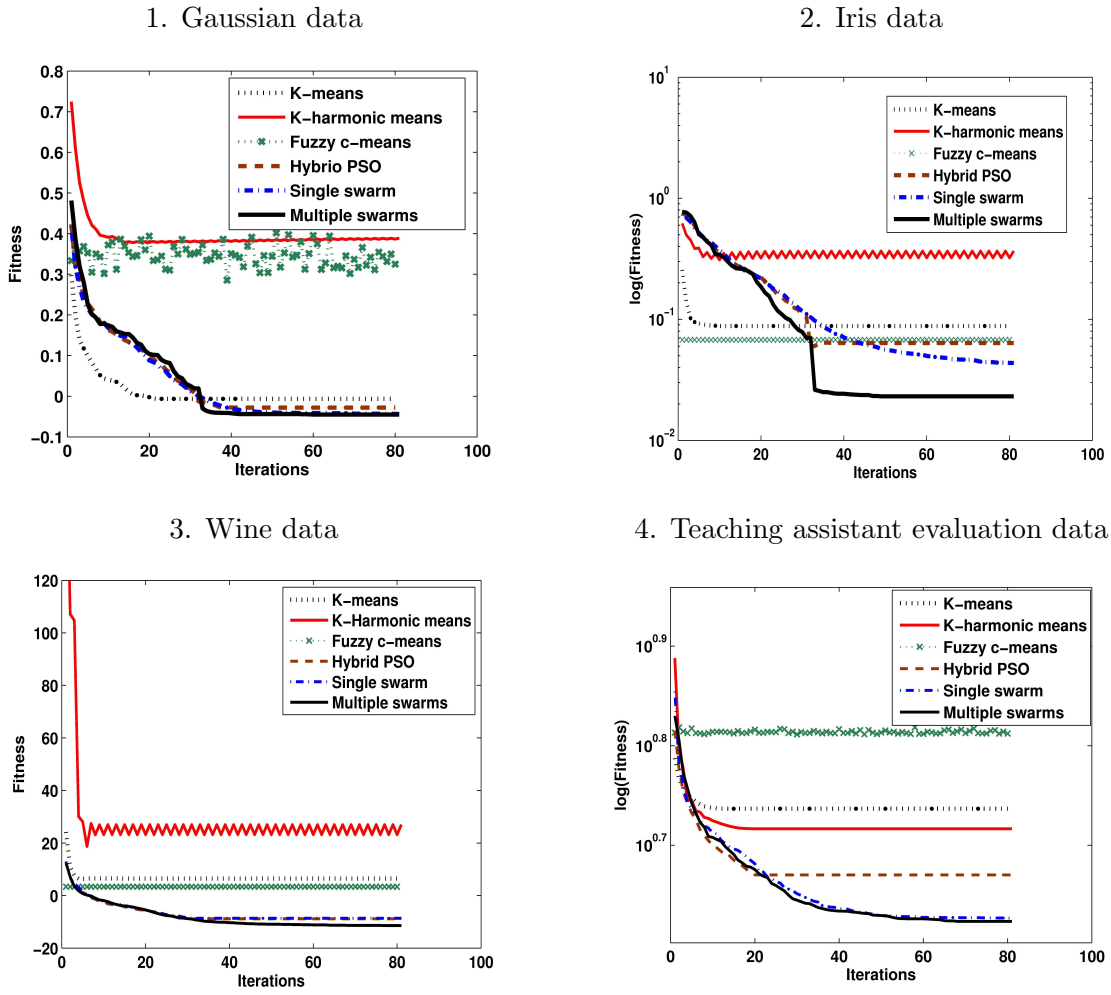
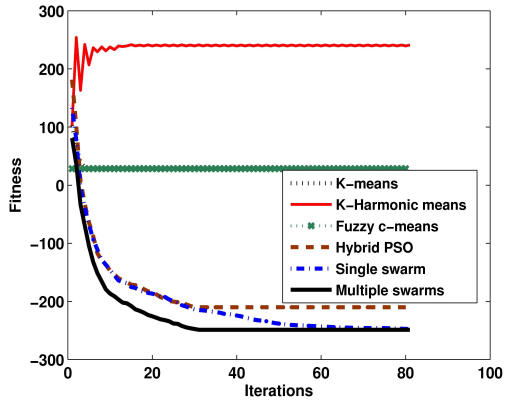


Figure 3.12: Comparing the convergence of the proposed multiple swarms clustering with other approaches in terms of combined measure for Gaussian, iris, wine and teaching assistant evaluation data sets

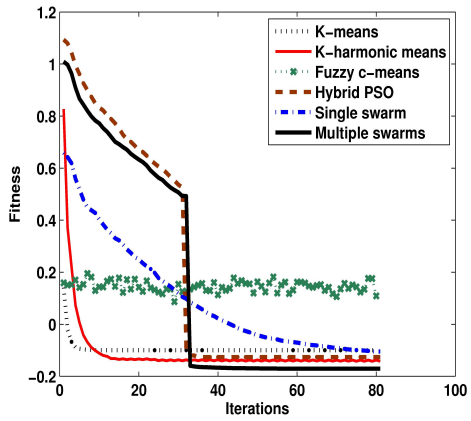
c-means, hybrid PSO and single swarm clustering through 80 iterations different data sets.

As illustrated in Fig. 3.12 and Fig. 3.13, the multiple swarms clustering approach can provide better solutions for the majority of data sets due to its strong and effective search ability, which confirms the analytical derivations provided in appendix **B**. A sudden drop in the fitness value of some data sets shows the beginning of the exploitation phase, where the cooperation between swarms starts. Also, K-means, K-harmonic means and fuzzy c-means clustering techniques converge quickly and they require less computational time to get to the final solution. Moreover, the multiple swarm approach outperforms single swarm and hybrid PSO approaches as it delegates a portion of the search space to each swarm. As compared to the single swarm technique, the proposed approach accelerates the convergence of the clustering task.

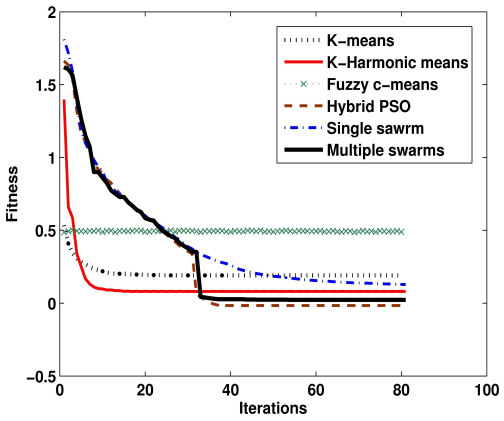
5. Breast cancer data



6. Zoo data



7. Glass identification data



8. Diabetes data

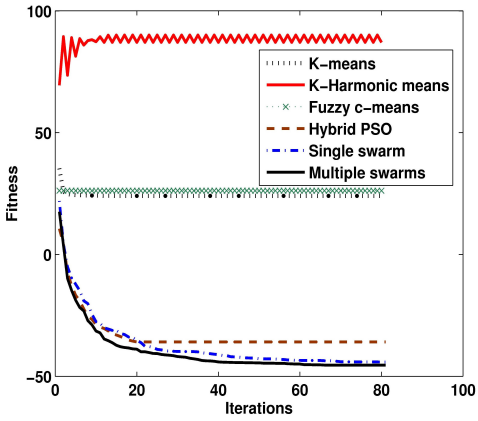


Figure 3.13: Comparing the convergence of the proposed multiple swarms clustering with other approaches in terms of combined measure for breast cancer, zoo, glass identification and diabetes data sets

Moreover, hybrid PSO may provide better solutions when the compactness measure is of interest since K-means algorithm is developed based on this measure. For the other measures such as separation and Turi's index, PSO can still yield proper initial solution to K-means. However, K-means algorithm may not be able to improve the provided results since it only considers compactness measure. The proposed cooperative swarms approach does not have this issue and it is able to deal with different types of measures in the objective function. We have also investigated the statistical significance of the difference between the proposed approach and the other methods using T-test at a significance level 5%. The detailed results are provided in appendix **C**.

3.4.2 Multiple swarms vs. single swarm as dimensionality of data increases

In this section, we examine the influence of the increasing dimensionality of data on the performance of multiple swarms and single swarm clustering approaches. The simulations are done using wine, breast cancer, zoo and glass identification data sets using the *combined measure*, and the results have been illustrated in Fig. 3.14. To obtain the results at a certain dimension d, the first d dimensions of the feature space are considered for the associated data set.

As can be seen from Fig. 3.14, the multiple swarms clustering approach outperforms the single swarm technique as the dimensionality of data increases, confirming the analytical derivations demonstrated in appendix **B**.

3.4.3 Multiple swarms vs. single swarm as number of clusters increases

In this section, the effect of the number of clusters on the performance of the proposed approach and the single swarm clustering is investigated. The simulations are performed using different data sets and the corresponding results are illustrated in Fig. 3.15 and Fig. 3.16.

As presented in Fig. 3.15 and Fig. 3.16, the multiple swarms clustering approach leads to a better outcome in terms of fitness value (here combined measure), as the number of clusters increases as derived analytically in appendix **B**.

3.4.4 High dimensions and large number of clusters

To study the performance of the proposed approach in higher dimensions and the existence of large number of clusters, three high dimensional Gaussian data sets are considered. First set denoted by High dimension D25 includes a total of 600 samples drawn from four 25-dimensional Gaussian classes, the second set denoted

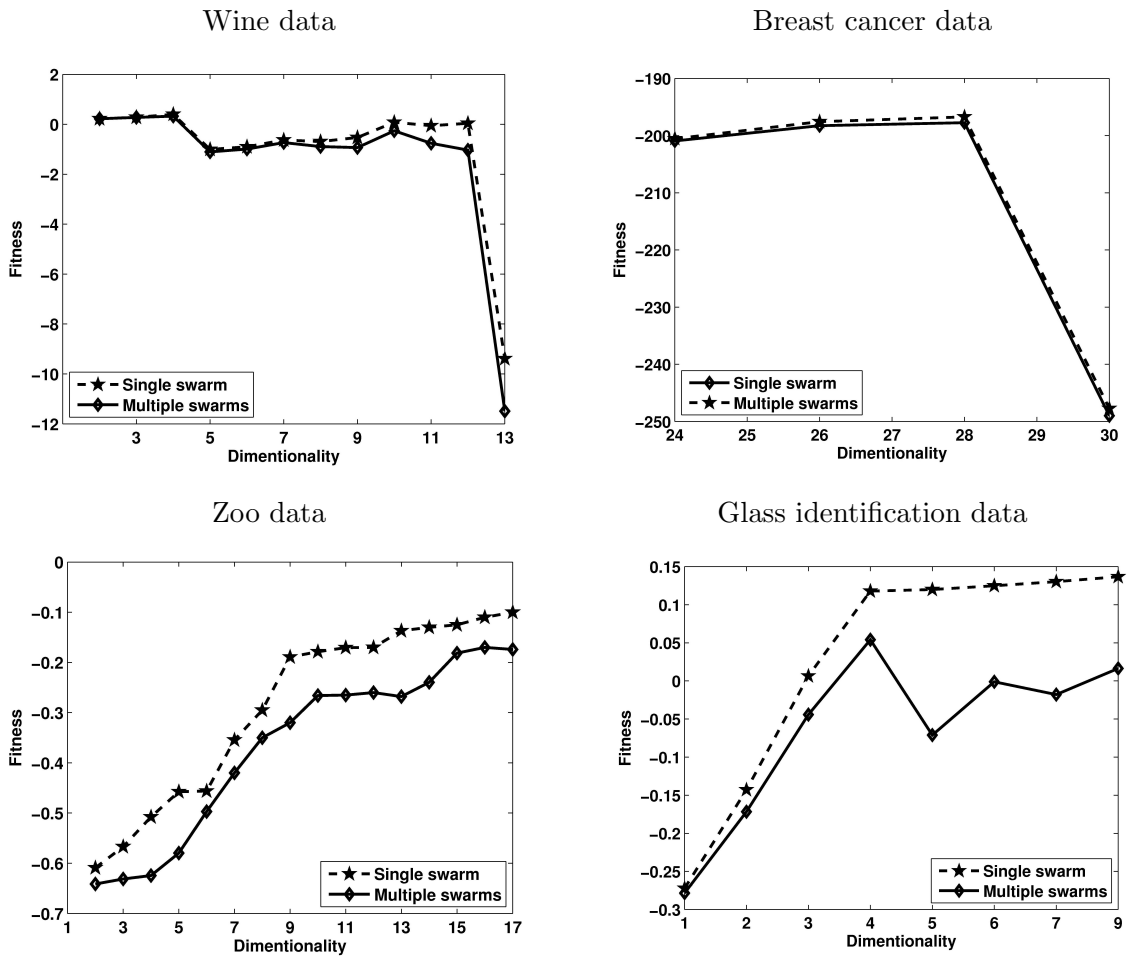


Figure 3.14: Comparing the performance of the single swarm and multiple swarms clustering approaches as dimensionality of data increases

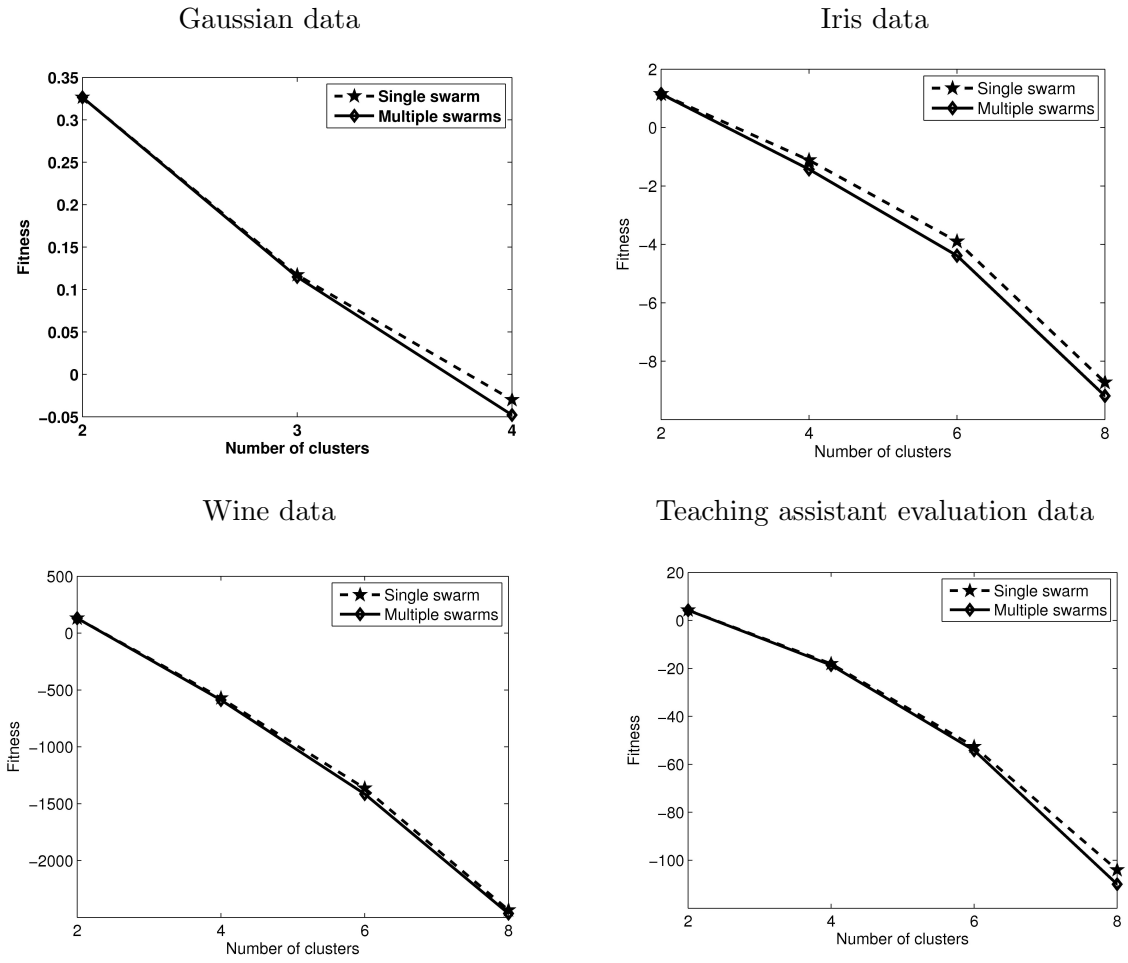


Figure 3.15: Comparing the performance of the single swarm and multiple swarms clustering approaches as the number of clusters increases: Gaussian, iris, wine and teaching assistant evaluation data sets

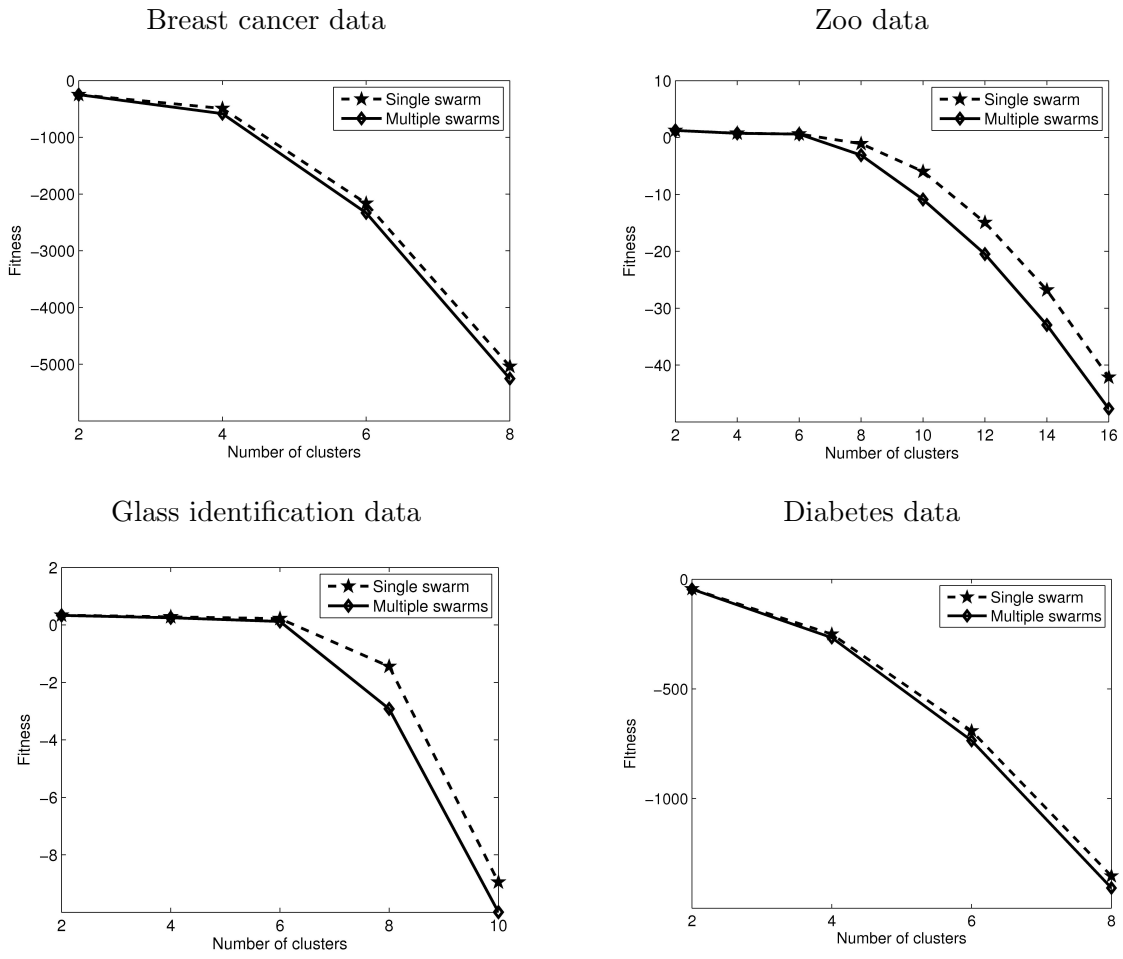


Figure 3.16: Comparing the performance of the single swarm and multiple swarms clustering approaches as the number of clusters increases: breast cancer, zoo, glass identification and diabetes data sets

Table 3.11: Average and standard deviation comparison of diffirent measures for high dimension D25 data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KH	3.4478[0.09]	-6.669[0.23]	2.002[0.09]	0.83[1.59]
KHM	3.39[0.0001]	-6.45[0.0001]	1.98[0.0001]	3.42[3.83]
FCM	3.44[0.03]	-5.94[0.036]	2.1[0.03]	1.24[1.39]
Hybrid PSO	3.4[0.07]	-6.65[0.058]	1.96[0.05]	1.116[1.41]
Single swarm	10.6[0.59]	$\mathbf{-88.17}$[2.47]	0.68[0.68]	-1.29[1.06]
Multiple swarms	**0.7**[0.86]	-84.83[2.74]	$\mathbf{-1.44}$[0.95]	$\mathbf{-1.3335}$[1.0813]

Table 3.12: Average and standard deviation comparison of diffirent measures for high dimension D25N25 data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
KH	65.4976[0.912]	-69.36[9.8]	46.23[2.05]	1.4[1.07]
KHM	75.6131[0.0001]	0.76[0.0001]	64.92[0.0001]	$0.75e07$[$0.85e07$]
FCM	75.3964[0.0001]	0.76[0.28]	64.73[0.0001]	$0.2004e9$[$0.0058e9$]
Hybrid PSO	66.285[0.8479]	-69.2[12.05]	47.23[1.65]	1.36[1.28]
Single swarm	181.7[15.5]	$\mathbf{-948}$[25]	77.16[14.49]	-1.368[0.87]
Multiple swarms	**11.682**[1.2341]	-918[68.5]	$\mathbf{-1.5}$[1.13]	$\mathbf{-1.59}$[1.56]

by High dimension N25D25 contains a total of 1250 samples sampled from twenty five 25-dimensional Gaussian classes and the third set denoted by High dimension N100D100 encompasses a total of 1250 samples generated from one hundred 100-dimensional Gaussian classes. The results for these data sets are provided in Tables 3.11-3.13.

3.5 Summary

In this chapter, a novel clustering approach based on multiple cooperative swarms was proposed. Using a super-swarm, the multiple cooperative swarms approach assigns a portion of the search space to each swarm. This strategy boosts its exploration ability, as each swarm deals with a part of the search space. Each

Table 3.13: Average and standard deviation comparison of diffirent measures for high dimension D100N100 data

Method	Compactness[σ]	Separation[σ]	Combined[σ]	Turi's index[σ]
Single swarm	$8.18e03$[29.9]	$-1.28e06$[$1.9e03$]	$-1.4e05$[349]	$2.06e04$[147.2]
Multiple swarms	**0.04e03**[29]	$\mathbf{-1.28e06}$[$2.9e03$]	$\mathbf{-1.4e05}$[513]	$\mathbf{-0.004e04}$[39]

swarm explores its own region while cooperating with other swarms; it knows the global best of other swarms and attempts to find a point whose cumulative distance from other clusters' centers is maximized. Each swarm also tends to minimize the within-cluster distance. The proposed multiple cooperative swarms clustering approach is applied to cluster eight sets of data. It outperforms the other methods because of distributing the search space among multiple swarms and using multiple cooperating swarms. The proposed clustering technique also facilitates clustering data with high dimensions and a large number of clusters.

Similar to most of the partitional clustering approaches, the multiple cooperative swarms approach needs to be provided a priori the number of clusters. This has always been a challenging task in the area of partitional clustering. In the following chapter, it is shown how stability analysis can be used to estimate the number of clusters for the underlying data using multiple cooperative swarms approach.

Chapter 4

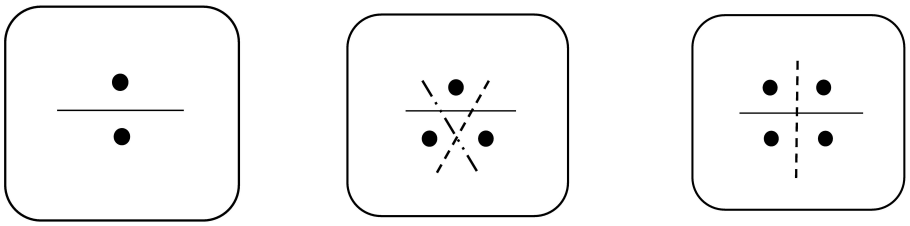
Stability-based Model Order Selection for Multiple Cooperative Swarms Clustering

Extracting different clusters of a given data is an appealing topic in swarm intelligence applications. In this chapter, a stability analysis is proposed to determine the model order of the underlying data using multiple cooperative swarms clustering. The mathematical explanations demonstrating why multiple cooperative swarms clustering leads to more stable and robust results than those of single swarm clustering are then provided. The proposed approach is evaluated using different data sets and its performance is compared with that of other clustering techniques.

4.1 Introduction

In data clustering, recognizing subgroups of the given data is of interest. A vast number of clustering techniques have been developed to deal with data based on different assumptions about the distribution, shape and size of the data. Most of the clustering techniques require a priori knowledge about the number of clusters [11], [12], whereas some other approaches are capable of extracting such information [47].

Swarm intelligence approaches such as particle swarm optimization, biologically inspired by the flocking behavior of birds [20], have been applied for clustering applications [2], [6], [8], [9], [23]. The goal of PSO-based clustering techniques is usually to find cluster centers. Most of the recent swarm clustering techniques use a single swarm approach to reach a final clustering solution [2], [3], [9]. Multiple swarms clustering has been recently proposed in [6]. The multiple swarms clustering approach is useful to deal with high dimensional data as it uses a *divide and conquer strategy*. In other words, it distributes the search space among multiple swarms, each of which explores its associated division while cooperating with others. The



(a) Stable (b) Unstable (c) Unstable

Figure 4.1: Examples of stable and unstable clustering when two clusters are desired.

novelty of this chapter is to apply the stability analysis for determining the number of clusters in underlying data using multiple cooperative swarms [48]. Also, we explain in this chapter why the proposed multiple cooperative approach is able to provide a more robust solution, in terms of mathematical demonstration, as compared with a single swarm approach.

In the following, a discussion on stability analysis for data clustering is provided.

4.2 Stability Analysis

Determining the number of clusters in data clustering is known as a model order selection problem. There exist two main stages in model order selection. First, a clustering algorithm should be chosen. Then, the model order needs to be extracted, given a set of data [47], [49].

Most of the clustering approaches assume that the model order is known in advance. Here, we employ stability analysis to obtain the number of clusters using a multiple cooperative swarms clustering approach. A description of stability analysis is provided before describing the core algorithm.

Stability concept is used to evaluate the robustness of a clustering algorithm. In other words, the stability measure indicates how much the results of the clustering algorithm are reproducible on other data drawn from the same source. Some examples of stable and unstable clustering are shown in Fig. 4.1 when the aim is to cluster the presented data into two groups.

As can be seen in Fig. 4.1, data points shown in Fig. 4.1.(a) provide a stable clustering solution in a sense that the same clustering results are obtained by repeating a clustering algorithm several times. However, the data points illustrated in Fig. 4.1.(b) and Fig. 4.1.(c) do not yield stable clustering solutions when two clusters are of interest. That is, different results are generated by running the clustering algorithm a number of times. Each line in Fig. 4.1 presents a possible clustering solution for the corresponding data. The reason for getting unstable clustering solutions in these cases is the inappropriate number of clusters. In other

words, stable results are obtained for these data sets by choosing a suitable number of clusters. The proper number of clusters for these data are three and four, respectively.

As a result, one of the issues that affects the stability of the solutions produced by a clustering algorithm is the model order. For example, by assuming a large number of clusters the algorithm generates random groups of data influenced by the changes observed in different samples. On the other hand, by choosing a very small number of clusters, the algorithm may compound separated structures together and return unstable clusters [47]. As a result, one can utilize the stability measure for estimating the model order of the unlabeled data [48].

4.3 Stability-based Model Order Selection

The multiple cooperative swarms clustering data requires a priori knowledge of the model order. In order to enable this approach to estimate the number of clusters, the stability approach is taken into consideration. We use the stability method introduced by Lange et al. [47] for the following reasons:

- it requires no information about the data being processed,
- it can be applied to any clustering algorithm,
- it returns the correct model order using the notion of maximal stability.

The required procedure for model order selection using stability analysis is provided in Algorithm **4.1**.

Regarding the stability-based model order selection algorithm, a number of issues should be explained as given next.

4.3.1 Classifier ϕ

A set of labeled data is required for training a classifier ϕ. The data set Y_1 and its clustering solution from algorithm A_k, i.e., $T_1 := A_k(Y_1)$, can be used to establish a classifier. There are a vast range of classifiers that can be used for classification. In this thesis, k-nearest neighbor (KNN) classifier was chosen as it requires no assumption on the distribution of data. Moreover, k is set to 25 for the k-nearest neighbor classifier.

4.3.2 Distance of solutions provided by clustering and classifier for the same data

Having a set of training data, the classifier can be tested using a test data Y_2. Its solution is represented by $T_2' = \phi(Y_2)$. But, there exists another solution for the

Algorithm 4.1 Model order selection using stability analysis

for $k \in [2, \cdots, K]$ **do**
  for $r \in [1, \cdots, rmax]$ **do**
    -Randomly split the given data Y into two halves Y_1, Y_2.
    -Cluster Y_1, Y_2 independently using an **appropriate clustering approach;** i.e., $T_1 := A_k(Y_1),\ T_2 := A_k(Y_2)$.
    -Use (Y_1, T_1) to train classifier $\phi(Y_1)$ and compute $T_2' = \phi(Y_2)$.
    -Calculate the distance of the two solutions T_2 and T_2' for Y_2; i.e., $d_r = d(T_2, T_2')$.
    -Again cluster Y_1, Y_2 by assigning random labels to points.
    -Extend *random clustering* as above, and obtain the distance of the solutions; i.e., dn_r.
  end for
  -Compute the stability $stab(k) = mean_r(d)$.
  -Compute the stability of random clusterings $stab_{rand}(k) = mean_r(dn)$.
  - $s(k) = \frac{stab(k)}{stab_{rand}(k)}$
end for
-Select the model order k^* such that $k^* = arg\min_k\{s(k)\}$.

same data obtained from the multiple cooperative swarms clustering technique, i.e., $T_2 := A_k(Y_2)$. The distance of these two solutions is calculated by

$$d(T_2, T_2') = arg\min_{\omega\in\rho_k} \sum_{i=1}^{N} \vartheta\{\omega(t_{2i}) \neq t'_{2i}\}, \tag{4.1}$$

where

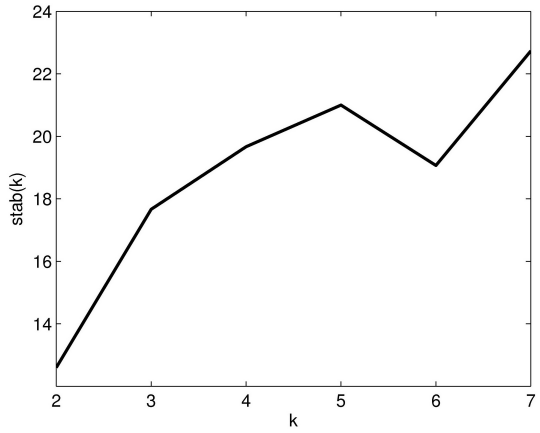
$$\vartheta\{t_{2i} \neq t'_{2i}\} = \begin{cases} 1 & \text{if } t_{2i} \neq t'_{2i}, \\ 0 & \text{otherwise}\cdot \end{cases} \tag{4.2}$$

Also, ρ_k contains all permutations of k labels and ω is the optimal permutation in ρ_k which produces the maximum agreement between two solutions [47].

4.3.3 Random clustering

The stability rate depends on the number of classes or clusters. For instance, the accuracy rate of 50% for binary classification is more or less the same as that of a random guess. However, this rate for $k = 10$ is much better than a random predictor. In other words, if a clustering approach outcomes the same accuracy for model orders k_1 and k_2, where $k_1 < k_2$, the clustering solution for k_2 is more reliable than the other solution. Hence, the primary stability measure obtained for a certain value k, $stab(k)$ in Algorithm **4.1**, should be normalized using a stability rate of a random clustering, $stab_{rand}(k)$ in Algorithm **4.1** [47]. The random clustering simply

(a) Excluding random clustering



(b) Including random clustering

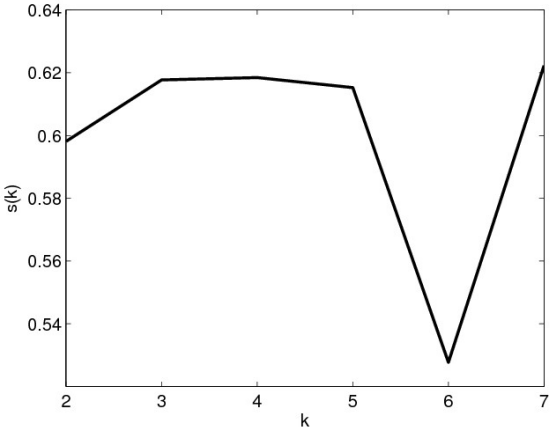


Figure 4.2: The effect of the random clustering on the selection of the model order.

means to assign a label between one and K to each data point randomly. The final stability measure for the model order k is obtained as follows:

$$s(k) = \{\frac{stab(k)}{stab_{rand}(k)}\}. \tag{4.3}$$

The effect of the random clustering is studied on the performance of the Zoo data set provided in section 4.5 to determine the model order of the data using K-means algorithm. The stability measure for different number of clusters with and without using random clustering is shown in Fig. 4.2.

As depicted in Fig. 4.2, the model order of the zoo data using K-means clustering is recognized as two without considering random clustering, while it becomes six, which is close to the true model order, by normalizing the primary stability measure to the stability of the random clustering.

4.3.4 Appropriate clustering approach

For a given data set, the algorithm does not provide the same result for multiple runs. Moreover, the model order is highly dependent on the type of **appropriate clustering approach** that is used in this algorithm (see Algorithm **4.1**), and there is no specific emphasis in the work of Lange et al. [47] on the type of clustering algorithm that should be used. K-means and K-harmonic means algorithms are either sensitive to the initial conditions or to the type of data. In other words, they cannot capture true underlying patterns of the data, and consequently the estimated model order is not robust. However, PSO-based clustering methods such as single swarm or multiple cooperative swarms clustering do not rely on initial conditions, and they are the search schemes which can explore the search space more effectively and can escape from local optimums. Further, as described in section 4.4, the multiple cooperative swarms clustering is more probable to get the optimal solution than the single swarm clustering and it can provide more stable and robust clustering solutions.

4.4 Stability Analysis: Multiple Swarms vs. Single Swarm

To analyze the stability of the single swarm and multiple swarms clustering, the probability of getting true cluster centers is studied in both approaches using the introduced stability-based scheme. In other words, to prove that the multiple cooperative swarms approach leads to more robust and stable results as compared to the single swarm, it is necessary to demonstrate that the probability of converging to true cluster centers using multiple cooperative swarms is greater than that of the single swarm clustering.

4.4.1 Probability of converging to an optimal clustering solution

First, let's study the probability of obtaining an optimal solution of the following optimization problem using particle swarm optimization

$$\begin{aligned} \mathcal{Z} = \min \ & \mathcal{F}(x) \\ \text{s.t.}: \quad & x \in \mathcal{S}, \end{aligned} \tag{4.4}$$

where S denotes the search space. Assume that $\mathcal{S}$ is a d-dimensional hyper-sphere of radius $\mathbf{r}$ and the optimal solution is located in a smaller d-dimensional hyper-sphere of radius r'. Accordingly, the probability of getting to the optimal solution by the PSO algorithm is given by

$$P_r = \frac{V(r',d)}{V(\mathbf{r},d)}, \tag{4.5}$$

where $V(r', d)$ and $V(\mathbf{r}, d)$ are volume of the d-dimensional hyper-spheres of radius r' and $\mathbf{r}$, respectively. The volume of the d-dimensional hyper-sphere of radius r is defined as

$$V(\mathbf{r}, d) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\mathbf{r}^d, \tag{4.6}$$

where $\Gamma(\cdot)$ stands for Gamma function. Considering equations (4.5) and (4.6), the probability of finding a solution in the optimal region is as follows:

$$P_r = (\frac{r'}{\mathbf{r}})^d. \tag{4.7}$$

In other words, the probability of finding an optimal solution decreases by increasing the dimensionality of data, provided $\mathbf{r}$ and r' do not change.

Now, consider a single swarm clustering whose true cluster centers are placed in K different d-dimensional hyper-spheres of radii r'_1, r'_2,$\cdots$,r'_K. To get an optimal solution, cluster centers should be selected from the associated regions. The probability of getting the cluster centers from the corresponding regions is given by

$$P_r^1 = \prod_{k=1}^{K} P_r(\mathbf{m}^k \in C^k), \tag{4.8}$$

where $P_r(\mathbf{m}^k \in C^k)$ indicates the probability of selecting the center of cluster k from its related region calculated by

$$P_r(\mathbf{m}^k \in C^k) = (\frac{r'_k}{\mathbf{r}})^d. \tag{4.9}$$

Using this expression, equation (4.8) can be rewritten as

$$P_r^1 = \prod_{k=1}^{K} (\frac{r'_k}{r})^d = \frac{(r'_1 r'_2 \cdots r'_K)^d}{\mathbf{r}^{d.K}}. \tag{4.10}$$

In the case of multiple swarms clustering, each swarm investigates a portion of the search space characterized by a d-dimensional hyper-sphere of radius r_k. Since $r_k < \mathbf{r}$ for all k, the following inequality is attained:

$$r_1 \cdots r_K < \mathbf{r}^K. \tag{4.11}$$

Because the dimensionality of the data is greater than one, inequality (4.11) is modified as

$$(r_1 \cdots r_K)^d < \mathbf{r}^{d.K}. \tag{4.12}$$

Assume the optimal solution for each swarm k in multiple swarms approach is situated in a d-dimensional hyper-sphere of radius r'_k. Accordingly, the probability of getting an optimal solution using multiple swarms at each iteration is calculated as

$$P_r^M = \prod_{k=1}^{K} P_r(\mathbf{m}^k \in C^k), \tag{4.13}$$

and it can be simplified as

$$P_r^M = \prod_{k=1}^{K} (\frac{r'_k}{r_k})^d = \frac{(r'_1 r'_2 \cdots r'_K)^d}{(r_1.r_2 \cdots r_K)^d}. \tag{4.14}$$

According to equations (4.10) and (4.14), the following output is obtained

$$\frac{P_r^M}{P_r^1} = \frac{\mathbf{r}^{d.K}}{(r_1.r_2 \cdots r_K)^d}. \tag{4.15}$$

Considering equations (4.12) and (4.15), one can simply observe that

$$\frac{P_r^M}{P_r^1} > 1, \quad or \ P_r^M > P_r^1. \tag{4.16}$$

Hence, the probability of obtaining the optimal solution using the multiple cooperative swarms clustering is greater than that of the single swarm clustering.

4.4.2 Stability of the proposed approach

Suppose that the underlying data Y is divided into two halves Y_1 and Y_2. Also, let A_K^1 and A_K^M denote the single swarm and multiple swarms clustering, respectively, where K indicates the model order.

The goal is to get to the true cluster centers, denoted by $(m^1, \cdots, m^K)$, for the given data Y. Let's have a look at the core idea of the algorithm **8** as depicted in Fig. 4.3.

According to Fig. 4.3, to prove that the multiple swarms clustering yields more stable and robust solutions than the single swarm clustering, it is required to show that

$$D^m(T_2^m, T_2^{'m}) < D^1(T_2^1, T_2^{'1}), \tag{4.17}$$

where T_2^m and T_2^1 indicate the released labels by multiple swarms and single swarm clusterings for data Y_2, respectively; and $D^m(\cdot)$ and $D^1(\cdot)$ are the distance measures obtained by multiple swarms and single swarm clustering, respectively.

Now, consider the labels T_2^m and $T_2^{'m}$ produced by multiple swarms clustering and the trained classifier $\phi(\cdot)$, respectively. The quality of the classifier's responses depends on the performance of the associated clustering approach, in this case multiple swarms clustering, on the data set Y_1. As a result, one can express the $D^m(T_2^m, T_2^{'m})$ in terms of the distance of the cluster centers obtained by the multiple swarms clustering using Y_1 and Y_2. The ultimate goal is to converge to the optimal cluster centers denoted by $(\mathbf{m}_1, \cdots, \mathbf{m}_K)$ by applying multiple cooperative swarms

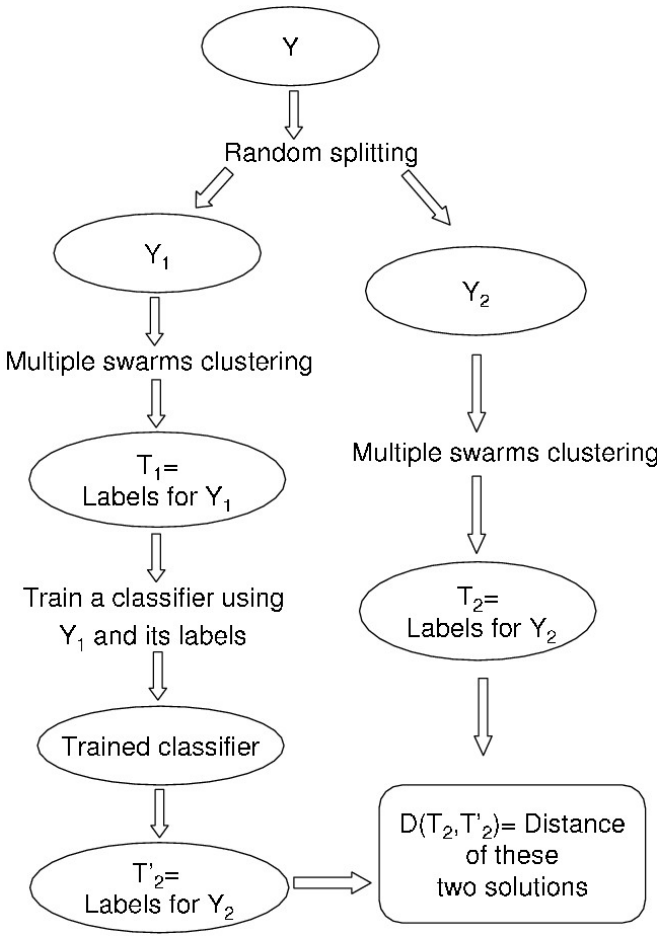


Figure 4.3: The core idea of the model order selection algorithm

clustering on both Y_1 and Y_2. In other words, $D^m(T_2^m, T_2'^m)$ is minimized when the multiple swarms clustering reaches the optimal solution using Y_1 and Y_2. The probability of converging to the optimal cluster centers by using the multiple swarms clustering on data sets Y_1 and Y_2, denoted by $P^m_{Y_1,Y_2}$, is given by

$$P^m_{Y_1,Y_2} = P^m_{Y_1} \cdot P^m_{Y_2}, \tag{4.18}$$

where $P^m_{Y_1}$ and $P^m_{Y_2}$ denote the probability of converging to the optimal solution by utilizing multiple swarms clustering on Y_1 and Y_2, respectively. In both cases, getting the same optimal solution $(\mathbf{m}_1, \cdots, \mathbf{m}_K)$ situated in K different d-dimensional hyper-spheres of radii r'_1, r'_2,$\cdots$,r'_K is of interest. However, the radius of the associated swarm regions of two different data sets may vary. Let $r_k^{(1)}$ and $r_k^{(2)}$ denote the radius of the swarm region k for data sets Y_1 and Y_2, respectively.

According to equation (4.14), $P^m_{Y_1,Y_2}$ is rewritten as

$$P^m_{Y_1,Y_2} = \prod_{k=1}^{K} (\frac{r'_k}{r_k^{(1)}})^d \cdot \prod_{k=1}^{K} (\frac{r'_k}{r_k^{(2)}})^d = \prod_{k=1}^{K} \frac{(r'_k)^{2d}}{(r_k^{(1)} \cdot r_k^{(2)})^d} \cdot \tag{4.19}$$

Similarly by using single swarm clustering, the probability of converging to the optimal cluster centers for data sets Y_1, Y_2 denoted by $P^1_{Y_1,Y_2}$ is given by

$$P^1_{Y_1,Y_2} = P^1_{Y_1} \cdot P^1_{Y_2}. \tag{4.20}$$

By substituting the corresponding expressions for $P^1_{Y_1}$ and $P^1_{Y_2}$ and doing some simplifications, the following result is attained

$$P^1_{Y_1,Y_2} = \prod_{k=1}^{K} \frac{(r'_k)^{2d}}{(\mathbf{r})^{2d}}. \tag{4.21}$$

Considering equations (4.19) and (4.21), the following expression is concluded

$$\frac{P^M_{Y_1,Y_2}}{P^1_{Y_1,Y_2}} = \prod_{k=1}^{K} \frac{(\mathbf{r})^{2d}}{(r^{(1)}_k \cdot r^{(2)}_k)^d} = \frac{(\mathbf{r})^{2Kd}}{\prod_{k=1}^{K} (r^{(1)}_k \cdot r^{(2)}_k)^d}. \tag{4.22}$$

Since $(r^{(1)}_1 \cdots r^{(1)}_K)^d < \mathbf{r}^{d.K}$ and $(r^{(2)}_1 \cdots r^{(2)}_K)^d < \mathbf{r}^{d.K}$, it is clear that

$$(\mathbf{r})^{2Kd} > \prod_{k=1}^{K} (r^{(1)}_k \cdot r^{(2)}_k)^d. \tag{4.23}$$

In other words,

$$P^M_{Y_1,Y_2} > P^1_{Y_1,Y_2}. \tag{4.24}$$

Hence, the multiple swarms clustering can produce more stable and robust results using the proposed approach, compared to single swarm clustering.

4.5 Assessment of the Model Order Selection Approach for Multiple Cooperative Swarms Clustering

The performance of the proposed approach is evaluated and compared with other approaches such as single swarm clustering, K-means and K-harmonic means clustering using eight different data sets, seven of which are selected from the UCI machine learning repository [46], and the last being a speech data set taken from the standard TIMIT corpus [50]. The name of data sets chosen from UCI machine learning repository, their associated number of classes, samples and dimensions are provided in Table 4.1.

Also, the speech data include four phonemes: /aa/, /ae/, /ay/ and /el/, from the TIMIT corpus. A total of 800 samples from these classes was selected, and twelve mel-frequency cepstral coefficients [51] have been considered as speech features.

Table 4.1: Data sets selected from UCI machine learning repository

Data set	classes	samples	dimensionality
Iris	3	150	4
Wine	3	178	13
Teaching assistant evaluation (TAE)	3	151	5
Breast cancer	2	569	30
Zoo	7	101	17
Glass identification	7	214	9
Diabetes	2	768	8

Table 4.2: Average and standard deviation comparison of diffirent measures for speech data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.8328[0.8167]	0.0789[0.0142]	3.3093[0.327]
K-harmonic means	$3.54e05$[$2.62e05$]	0.0769[0.0001]	3.3242[0.0001]
Single swarm	-1.4539[0.8788]	**0.1098**[0.014]	1.5531[0.0372]
Cooperative swarms	$\mathbf{-1.6345}$[1.0694]	0.1008[0.0153]	**1.583**[0.0388]

The performance of the multiple cooperative swarms clustering approach is compared with K-means and single swarm clustering techniques in terms of Turi's validity index over 80 iterations (Fig. 4.4 and Fig. 4.5). The results are obtained by repeating the algorithms 30 independent times. For these experiments, the parameters are set as $w = 1.2$, $c_1 = 1.49$, $c_2 = 1.49$, $n = 30$ (for all swarms). In addition, the model order is considered to be equal to the number of classes.

As illustrated in Fig. 4.4, multiple cooperative swarms clustering provides better results as compared with K-means, as well as single swarm clustering approaches, in terms of Turi's index for a majority of the data sets.

In Tables 4.2-4.9, the multiple cooperative swarms clustering is compared with other clustering approaches using different cluster validity measures over 30 independent runs. The results presented for different data sets are in terms of average and standard deviation ([σ]) values.

As observed in Tables 4.2-4.9, multiple swarms clustering is able to provide better results in terms of the different cluster validity measures for most of the data sets. This is because it is capable of manipulating multiple-objective problems, in contrast to K-means (KM) and K-harmonic means (KHM) clustering, and it distributes the search space between multiple swarms and solves the problem more effectively.

Now, the stability-based approach for model order selection in multiple cooperative swarms clustering is studied. The PSO parameters are kept the same as before, and $rmax = 30$ and k is considered to be 25 for KNN classifier. The sta-

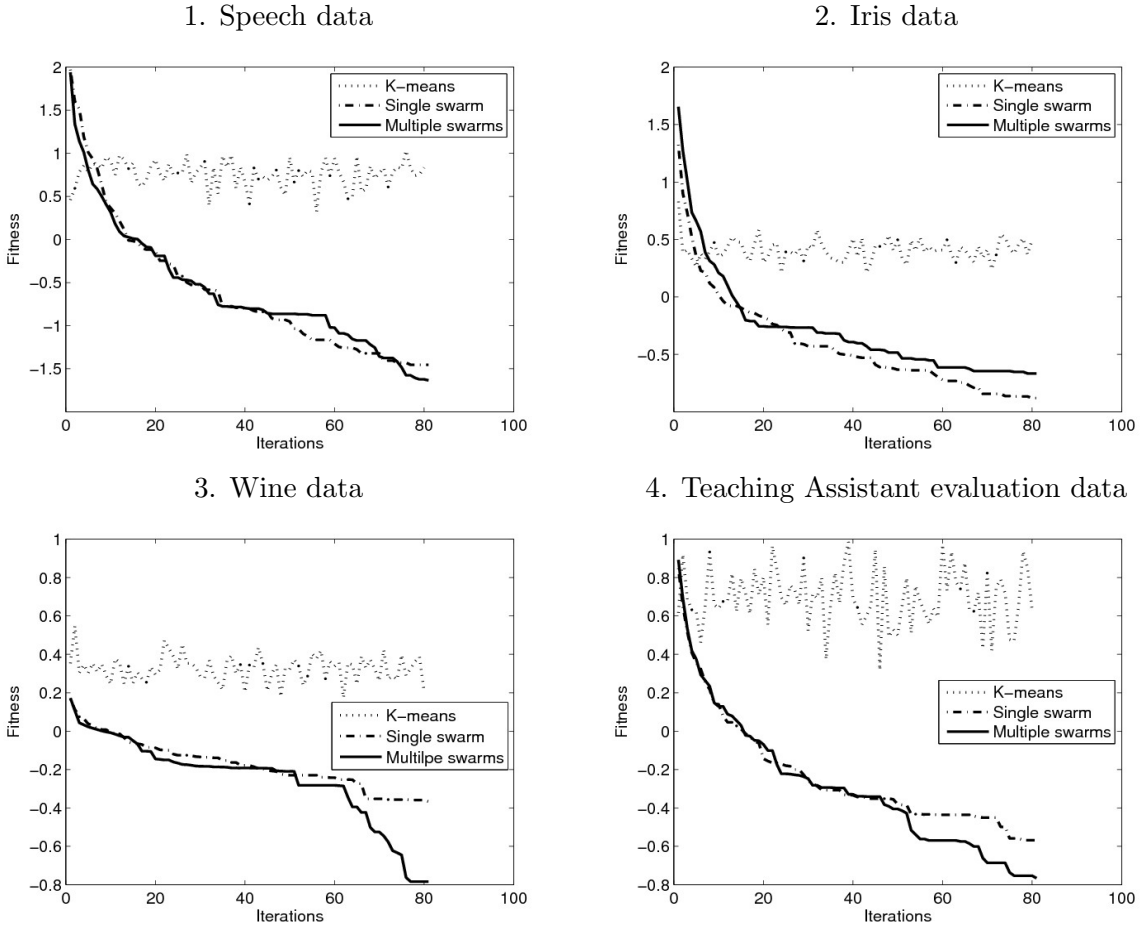


Figure 4.4: Comparing the performance of the multiple cooperative swarms clustering with K-means and single swarm clustering in terms of Turi's index: speech, iris, wine and TAE data sets

Table 4.3: Average and standard deviation comparison of diffirent measures for iris data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.4942[0.3227]	0.1008[0.0138]	3.0714[0.2383]
K-harmonic means	0.82e05[0.95e05]	0.0921[0.0214]	3.0993[0.0001]
Single swarm	−0.8802[0.4415]	**0.3979**[0.0001]	1.4902[0.0148]
Cooperative swarms	−**0.89**[1.0164]	**0.3979**[0.0001]	**1.48**[0.008]

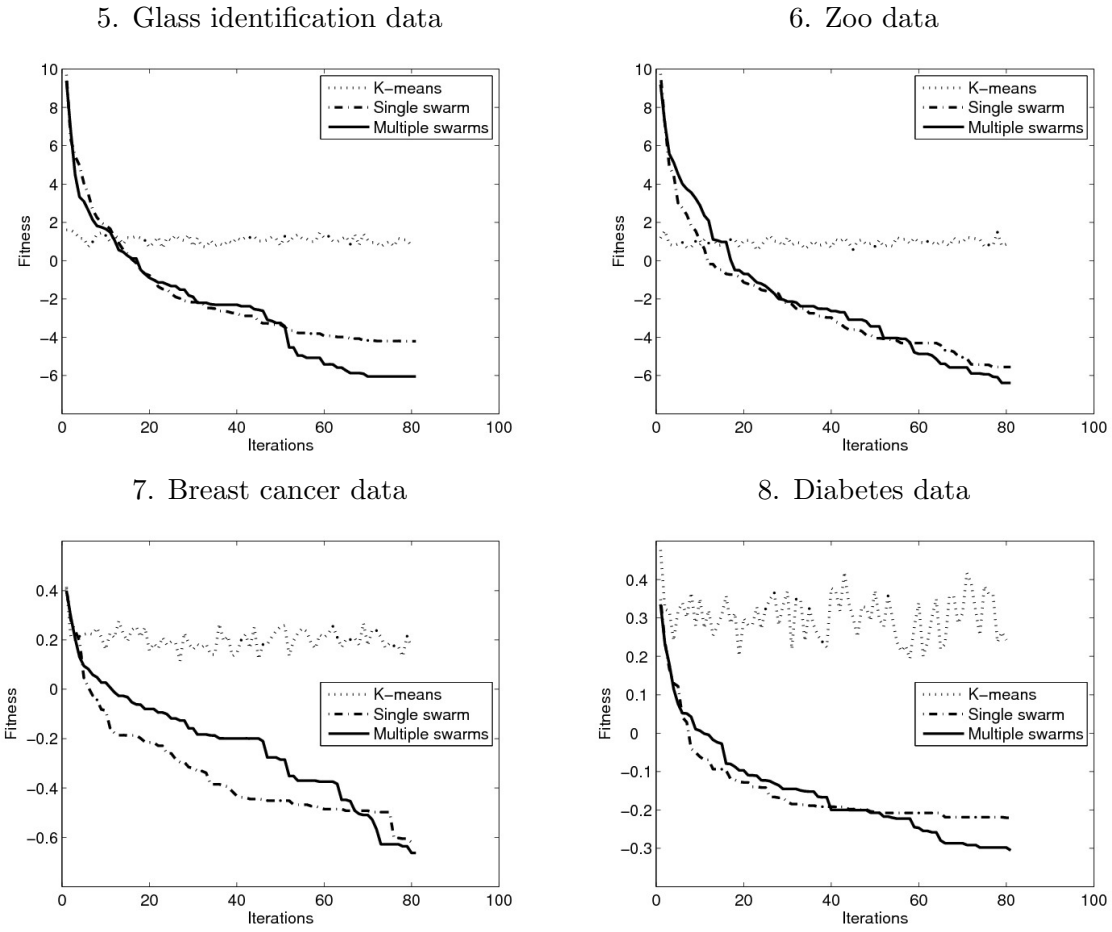


Figure 4.5: Comparing the performance of the multiple cooperative swarms clustering with K-means and single swarm clustering in terms of Turi's index: glass identification, zoo, breast cancer and diabetes data sets

Table 4.4: Average and standard deviation comparison of diffirent measures for wine data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.2101[0.3565]	0.016[0.006]	3.1239[0.4139]
K-harmonic means	2.83e07[2.82e07]	**190.2**[320.75]	2.1401[0.0149]
Single swarm	−0.3669[0.4735]	0.1122[0.0213]	1.3843[0.0026]
Cooperative swarms	**−0.7832**[0.8564]	0.0848[0.009]	**1.3829**[0.0044]

Table 4.5: Average and standard deviation comparison of diffirent measures for teaching assistant evaluation data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.6329[0.7866]	0.0802[0.0306]	3.2321[0.5205]
K-harmonic means	$1.36e06$[$1.23e06$]	0.123[0.0001]	2.7483[0.0001]
Single swarm	-0.5675[0.6525]	**0.1887**[0.0001]	1.4679[0.0052]
Cooperative swarms	$-$**0.7661**[0.7196]	**0.1887**[0.0001]	**1.4672**[0.004]

Table 4.6: Average and standard deviation comparison of diffirent measures for breast cancer data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.1711[0.1996]	0.0173[0.0001]	2.1768[0.0001]
K-harmonic means	0.88[0.95]	7.0664[38.519]	1.8574[0.0203]
Single swarm	-0.62[0.7997]	217.59[79.079]	1.7454[0.079]
Cooperative swarms	$-$**0.6632**[0.654]	**245.4857**[53.384]	**1.7169**[0.0925]

Table 4.7: Average and standard deviation comparison of diffirent measures for zoo data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.8513[1.0624]	0.2228[0.0581]	2.5181[0.2848]
K-harmonic means	1.239[1.5692]	0.3168[0.0938]	2.3048[0.1174]
Single swarm	-5.5567[3.6787]	**0.5427**[0.0165]	**2.0528**[0.0142]
Cooperative swarms	$-$**6.385**[4.6226]	0.5207[0.0407]	2.0767[0.025]

Table 4.8: Average and standard deviation comparison of diffirent measures for glass identification data

Method	Turi's index	Dunn's index	S_Dbw
K-means	0.7572[0.9624]	0.0286[0.001]	2.599[0.2571]
K-harmonic means	$0.89e05$[$1.01e05$]	0.0455[0.0012]	**2.0941**[0.0981]
Single swarm	-4.214[3.0376]	0.1877[0.0363]	2.6797[0.3372]
Cooperative swarms	$-$**6.0543**[4.5113]	**0.225**[0.1034]	2.484[0.1911]

Table 4.9: Average and standard deviation comparison of diffirent measures for diabetes data

Method	Turi's index[σ]	Dunn's index[σ]	S_Dbw[σ]
K-means	0.243[0.3398]	0.0137[0.0001]	2.297[0.0001]
K-harmonic means	1.88e07[1.9e07]	153.68[398.42]	2.0191[0.353]
Single swarm	−0.2203[0.2621]	**1298.1**[0.0001]	1.5202[0.027]
Cooperative swarms	−**0.3053**[0.3036]	**1298.1**[0.0001]	**1.5119**[0.0043]

bility measures of different model orders for the multiple cooperative swarms and other clustering approaches using different data sets are presented in Fig. 4.6 - Fig. 4.13. In these figures, k and $s(k)$ indicate model order and stability measure for the given model order k, respectively. Furthermore, the corresponding curves for single swarm and multiple swarms clustering approaches are obtained using Turi's validity index.

According to Fig. 4.6 - Fig. 4.13, the proposed approach using multiple cooperative swarms clustering is able to identify the correct model order for most of the data sets. Moreover, the best model order for different data sets can be obtained as provided in Table 4.10. The minimum value for stability measure given any clustering approach is considered as the best model order (k^*); i.e.,

$$k^* = arg\min_k\{s(k)\}\cdot \tag{4.25}$$

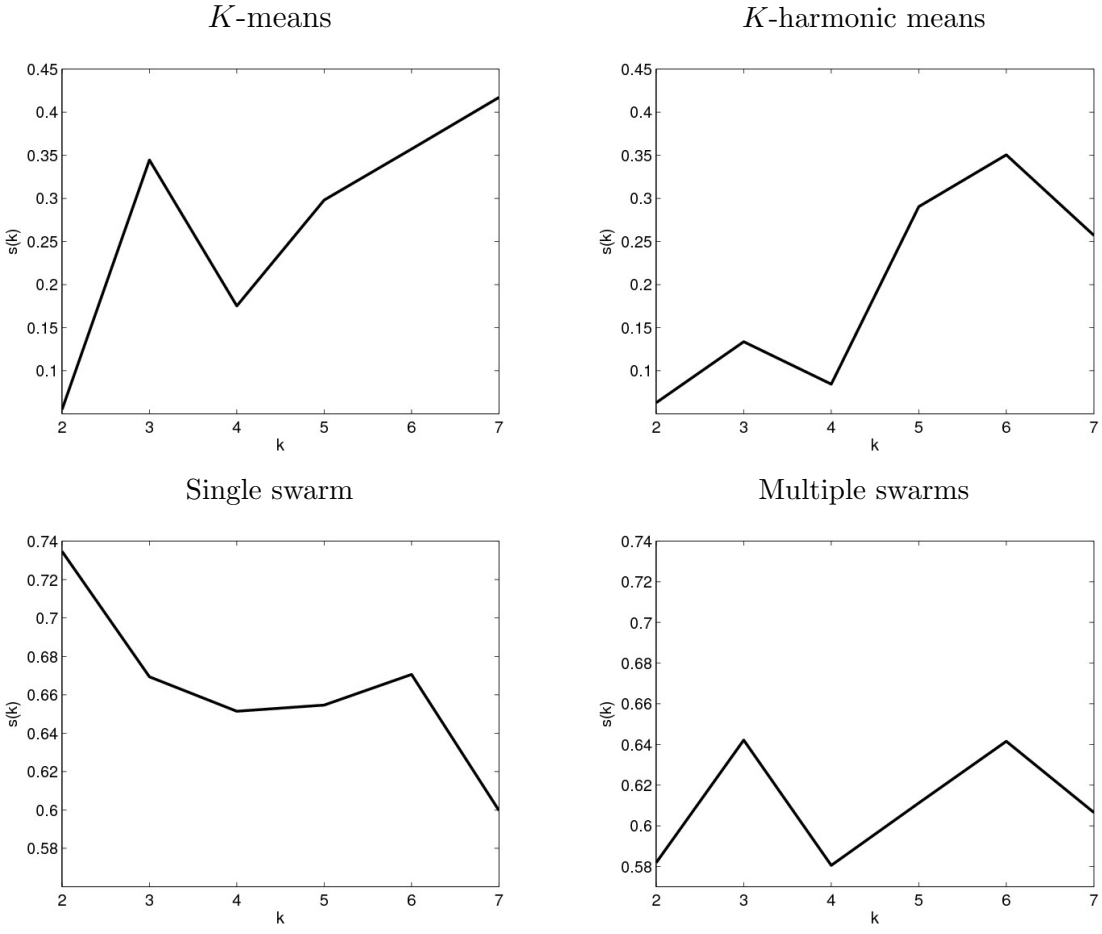


Figure 4.6: Stability measure as a function of model order: speech data

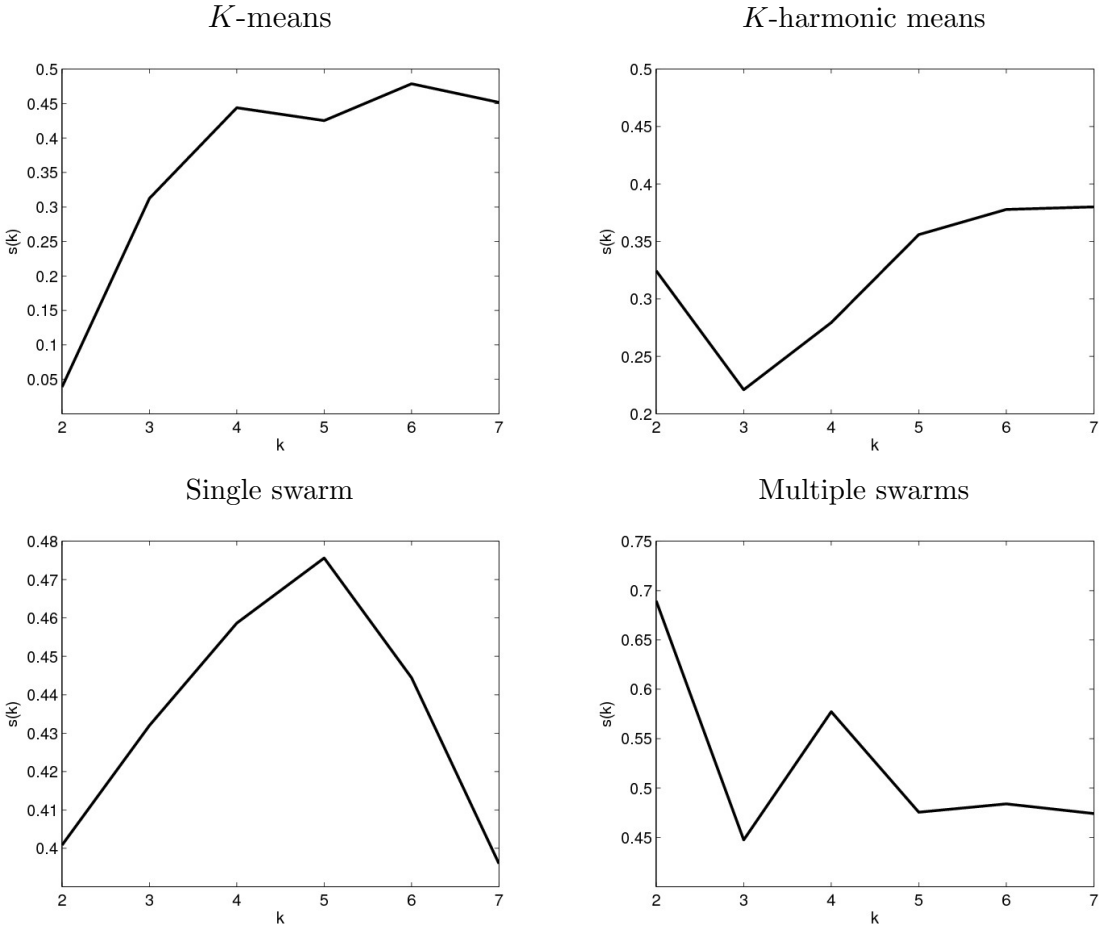


Figure 4.7: Stability measure as a function of model order: iris data

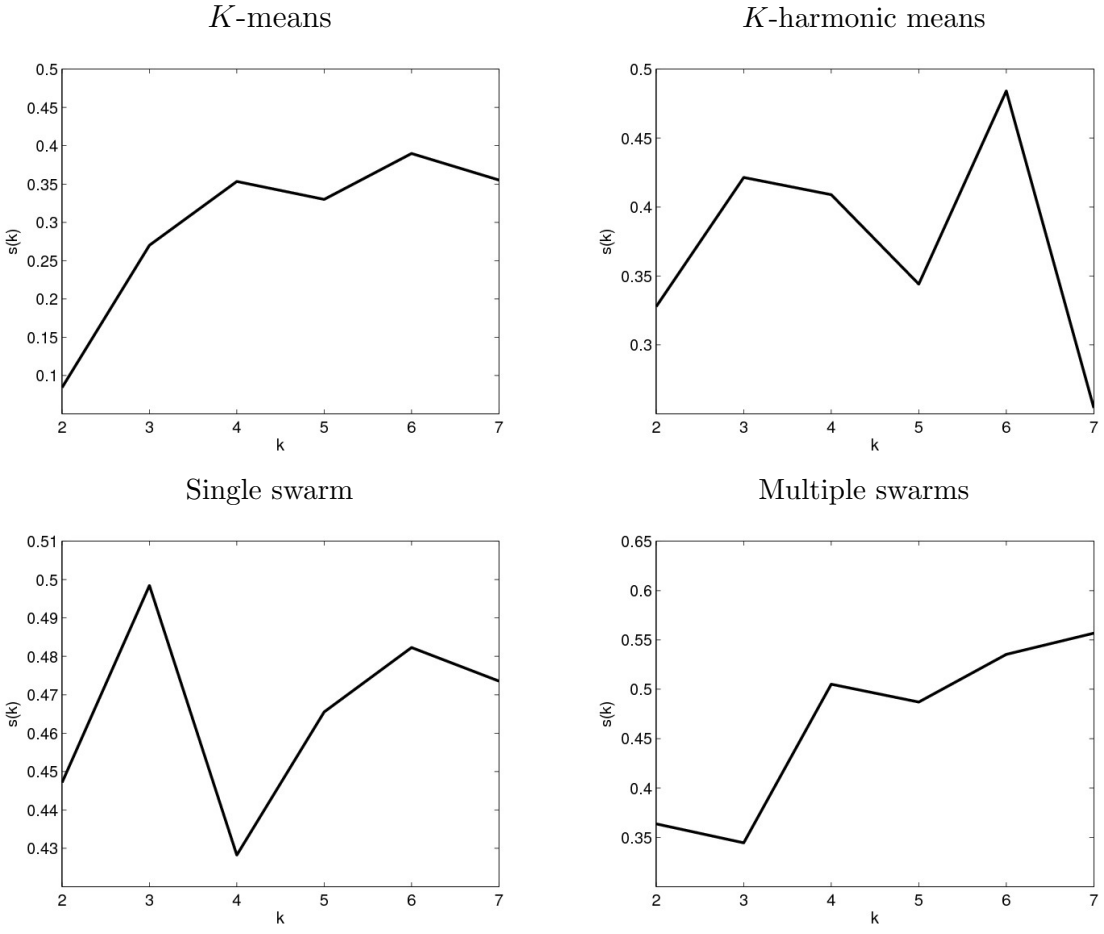


Figure 4.8: Stability measure as a function of model order: wine data

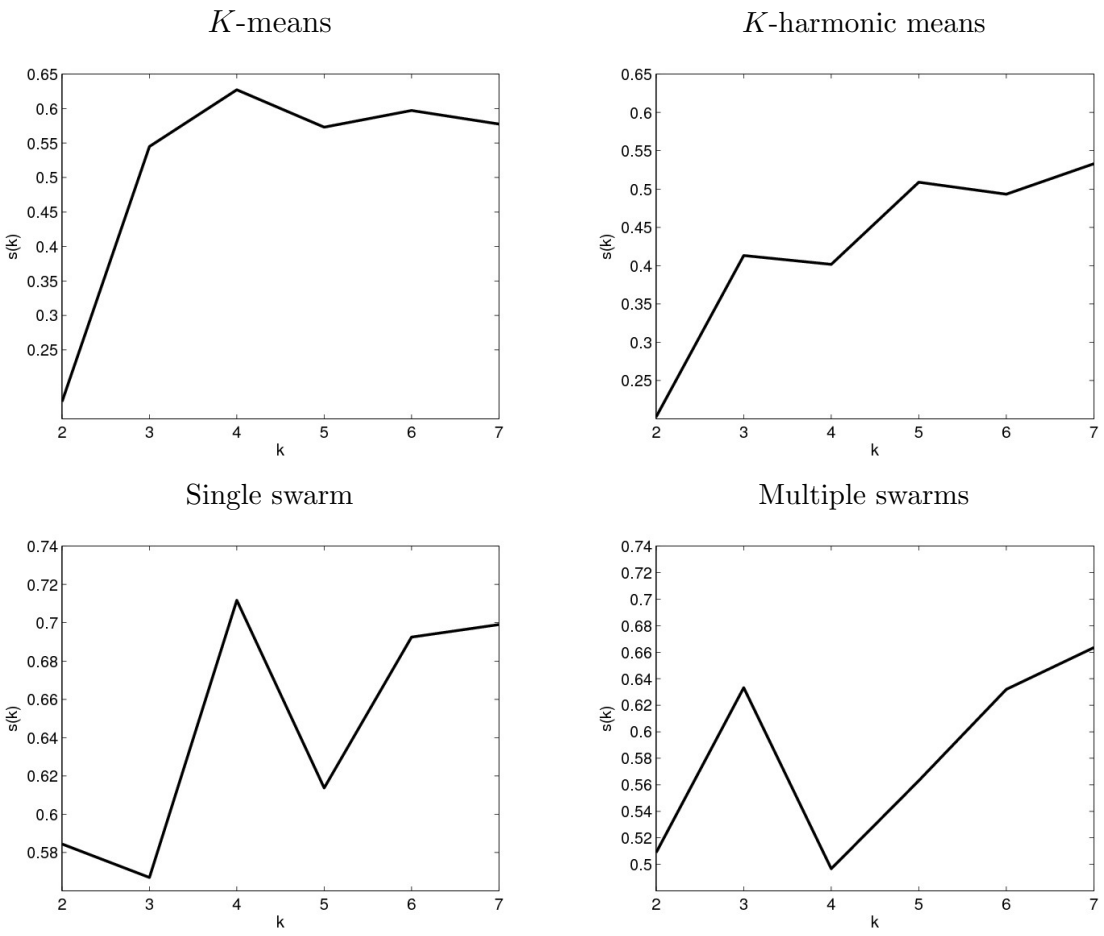


Figure 4.9: Stability measure as a function of model order: teaching assistant evaluation data

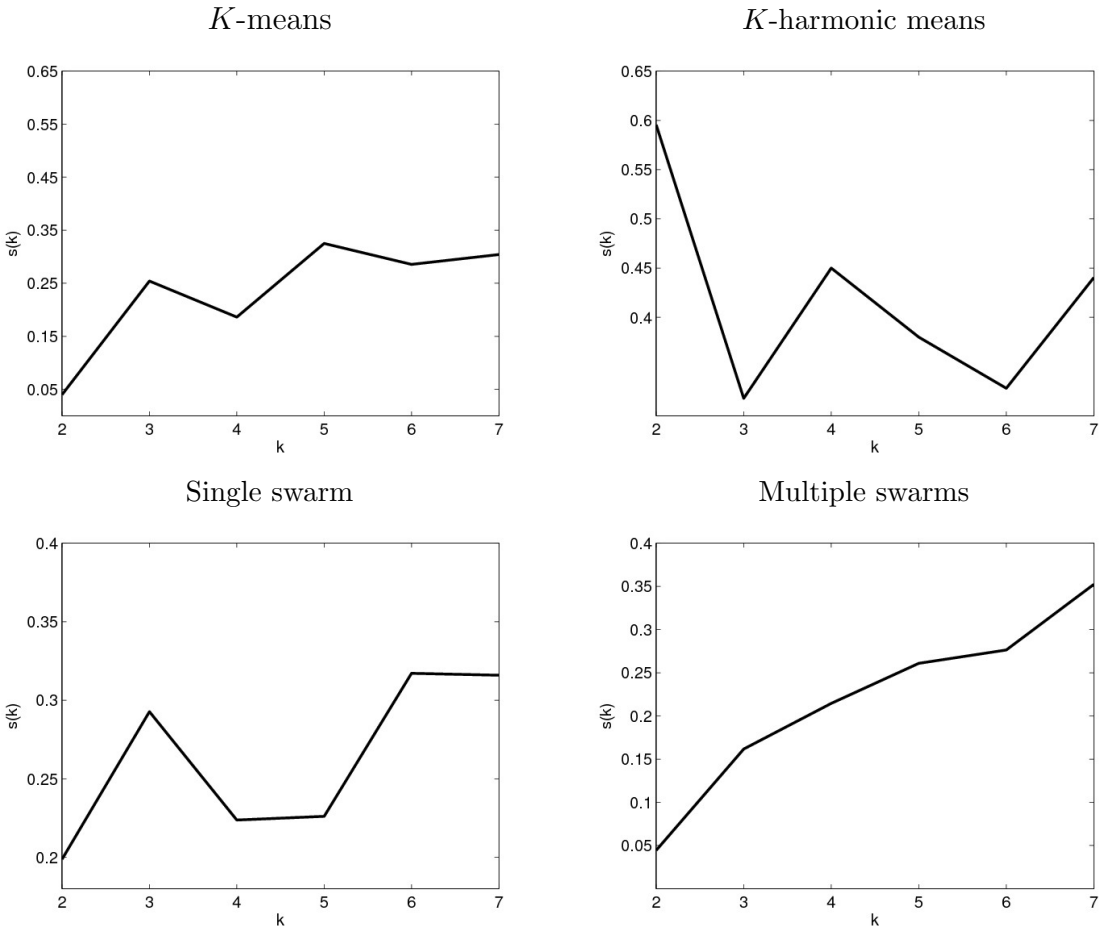


Figure 4.10: Stability measure as a function of model order: breast cancer data

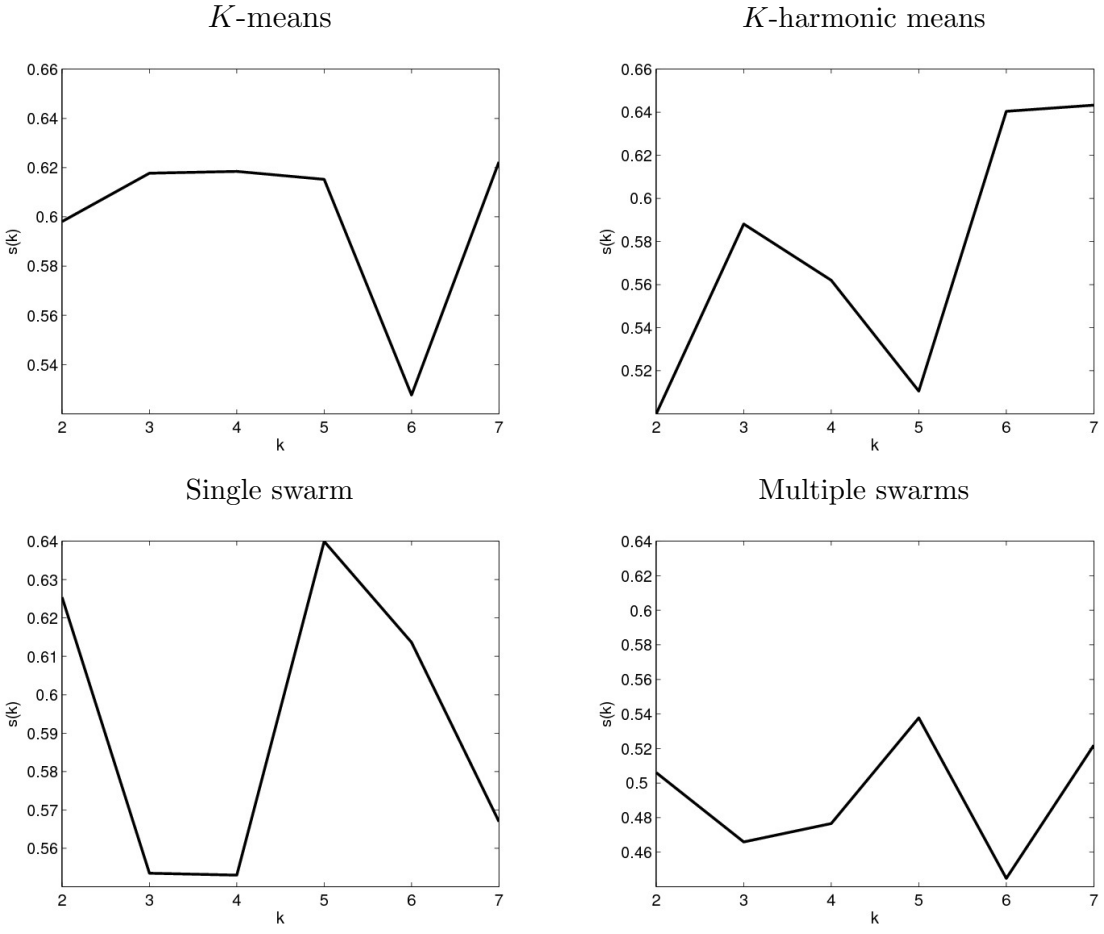


Figure 4.11: Stability measure as a function of model order: zoo data

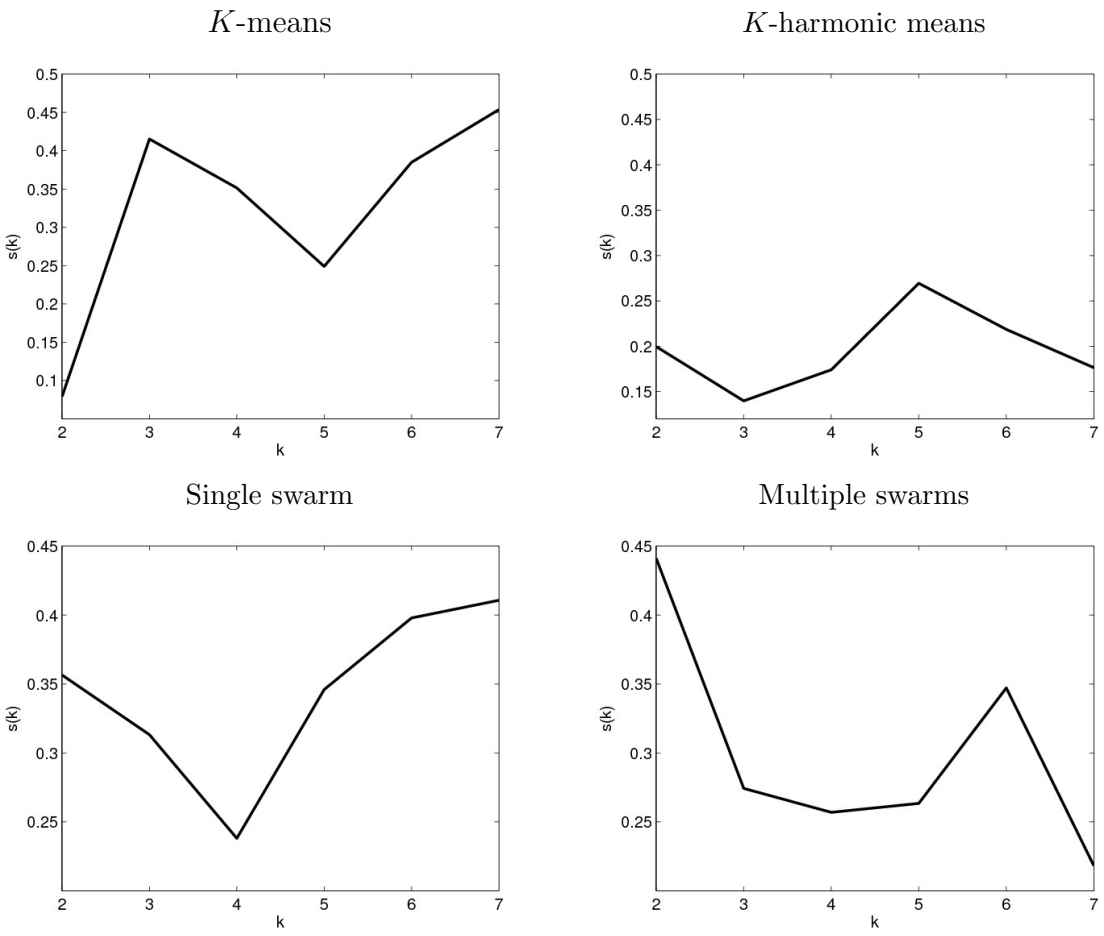


Figure 4.12: Stability measure as a function of model order: glass identification data

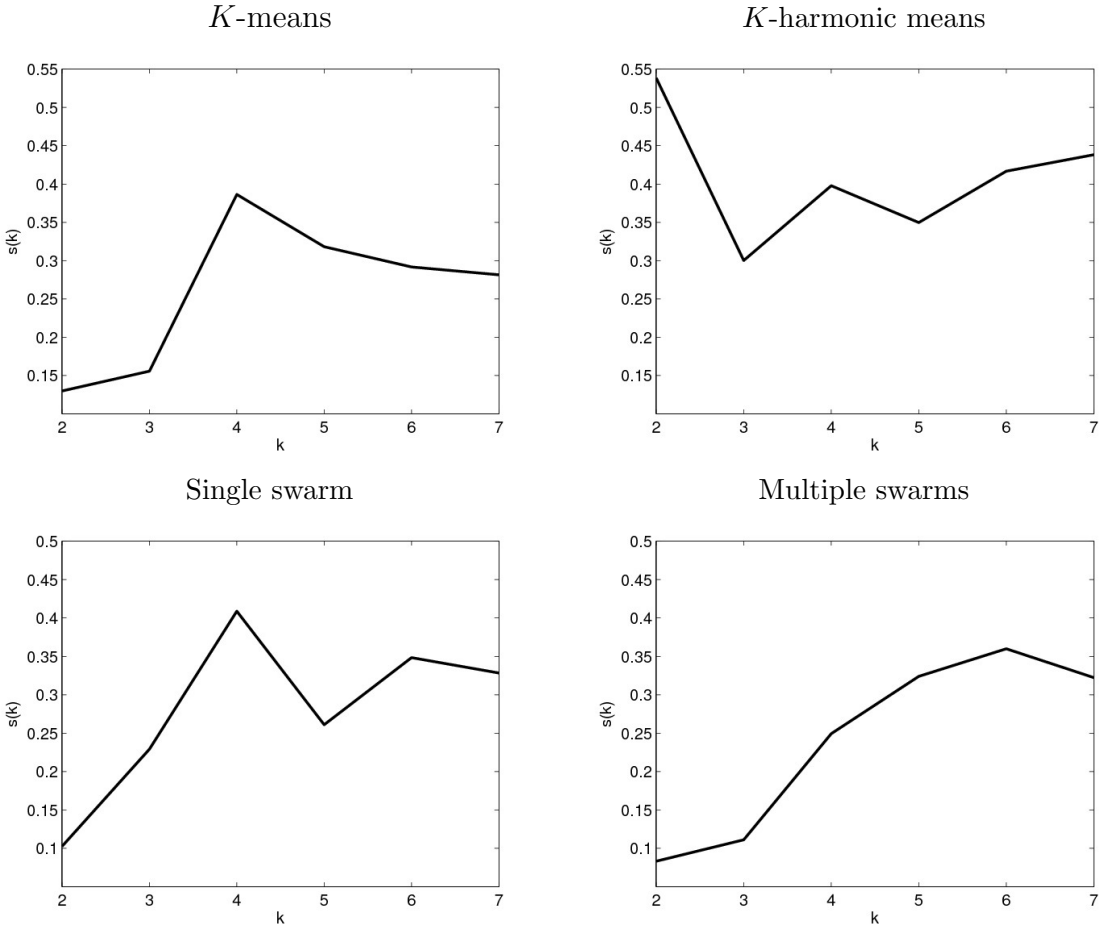


Figure 4.13: Stability measure as a function of model order: diabetes data

Table 4.10: The best model order (k^*) for data sets

Data set	KM	KHM	Single swarm			Multiple swarms		
			Turi	Dunn	S_Dbw	Turi	Dunn	S_Dbw
Speech	2	2	7	2	2	**4**	2	2
Iris	2	**3**	7	2	2	**3**	4	2
Wine	2	7	4	4	2	**3**	5	**3**
TAE	2	2	**3**	2	2	4	2	2
Breast cancer	**2**	3	**2**	**2**	**2**	**2**	**2**	**2**
Zoo	**6**	2	4	2	2	**6**	2	2
Glass identification	2	3	4	2	2	**7**	2	2
Diabetes	**2**	3	**2**	**2**	**2**	**2**	**2**	**2**

As presented in Table 4.10, K-means and K-harmonic means clustering approaches do not converge to the true model order using the stability-based approach for the most of the data sets. The performance of the single swarm clustering is partially better than that of K-means and K-harmonic means clustering because it does not depend on initial conditions and can escape trapping in local optimal solutions. Moreover, the multiple cooperative swarms approach using Turi's index provides the true model order for majority of the data sets. As a result, Turi's validity index is appropriate for the model order selection using the proposed clustering approach. Its performance, based on Dunn's index and S_Dbw index, is also considerable as compared to the other clustering approaches. Consequently, the proposed multiple cooperative swarms can provide better estimates for model order, as well as stable clustering results as compared to the other clustering techniques by using the introduced stability-based approach.

4.6 Summary

In this chapter, the stability analysis-based approach was introduced to estimate the model order of data using the multiple cooperative swarms clustering. We proposed to use the multiple cooperative swarms clustering to find the model order of the data, due to its robustness and stable solutions. Moreover, it has been shown that the probability of providing an optimal solution by multiple cooperative swarms clustering is higher than that of a single swarm scenario. To demonstrate the scalability of the proposed algorithm, it has been evaluated using eight different data sets. Its performance has also been compared with other clustering approaches.

In the following chapter, the application of the multiple cooperative swarms clustering approach in phoneme recognition is provided. The proposed swarm intelligence-based clustering approach is employed to build a modular classifier used for phoneme recognition.

Chapter 5

Application of the Proposed Multiple Cooperative Swarms Clustering for Phoneme Recognition

In this chapter, the approach we have developed is being applied to one of the most critical problems in the field of signal processing and speech recognition: phoneme recognition. First, the multiple cooperative swarms approach for clustering speech data is presented. Multiple cooperative swarms of flocking birds are employed to investigate the cluster centers of the given speech data from the standard TIMIT corpus. Moreover, this cooperative approach is applied to divide the phoneme recognition task into different subtasks in a modular classifier. The experiments indicate that using cooperative swarms clustering boosts the accuracy of the modular approach for phoneme recognition considerably.

5.1 Introduction

Spoken language processing has recently attracted a great deal of interest due to emerging multimedia, web, and audio mining applications. There exist lots of speech data, derived from different acoustic environments, various microphone characteristics and different speaking styles, which are required to be clustered into a limited number of groups. Thus, using a streamlined clustering technique is a crucial issue for automatic speech recognition (ASR) systems. The clustering technique can be applied to establish the architecture of ASR systems; for example, in modular approach for speech recognition, a clustering technique is required to decompose the recognition task into several subtasks [52],[53]. In a Gaussian mixture models approach for speech recognition, the centers of mixtures are also estimated using a clustering technique. Moreover, when using radial basis function

networks for phoneme recognition, the centers of basis functions are determined using a clustering technique [54].

Common clustering algorithms such as K-means and fuzzy c-means apply randomly generated points as the initial centers and update the position of the centers at every iteration. They only search a narrow neighborhood of the initial centers and may converge to local optimal solutions. In addition, when the dimensionality of data and the number of clusters increase, the search space expands exponentially, and consequently the clustering task becomes even more intractable. In speech recognition problems, the dimensionality of data and the possible number of clusters are relatively high, therefore there is a need for a more competent clustering technique. Swarm intelligence can be beneficial in this regard due to its population-based search mechanism.

Swarm intelligence approaches have been studied recently with a great deal of interest to tackle engineering problems. Particle swarm optimization (PSO) as a branch of swarm intelligence imitates the swarming behavior of flocks of birds [29]. PSO algorithm employs a swarm of individuals named particles to solve an optimization problem. It starts from an initial population and explores the search space by a number of iterations to reach a near-optimal solution. PSO has been applied successfully to several clustering applications such as gene clustering, document clustering and image segmentation [2], [3], [4], [9], [5]. As compared to conventional clustering techniques such as K-means, particle swarm clustering approaches are less sensitive to the effect of the initial conditions because of their population-based nature. Thus, it is more probable that particle swarm clustering yields near-optimal solutions.

There are two main categories of particle swarm clustering approaches: single swarm and multiple swarms. In single swarm clustering, an individual swarm explores the search space [3], [4], [8], [5]. This approach is effective particularly for problems with low dimensional data and a limited number of clusters. To deal with a high dimensional data and a large number of clusters, as happens in speech recognition problems, the multiple cooperative swarms can be used to handle the clustering task [6], [55], [56]. This approach distributes the search space among multiple swarms and each swarm is responsible for exploring a part of the search space, while interacting with other swarms to obtain the final clustering solution.

Here we study the application of a multiple cooperative swarms approach for phoneme data clustering as well as task decomposition at modular approach for phoneme recognition. The performance of the multiple cooperative swarms clustering approach is compared with other clustering techniques such as K-means, K-harmonic means, fuzzy c-means and single swarm clustering.

5.2 Speech Recognition

Automatic speech recognition is a process by which a machine identifies speech. It takes a human utterance as an input and returns the sequence of words or phrases as output. One simple method to recognize speech is a bottom-up approach, in which a speech waveform is first encoded into a sequence of phonemes. These phonemes are then mapped into a sequence of words [57], [58], [59] as illustrated in Fig. 5.1.

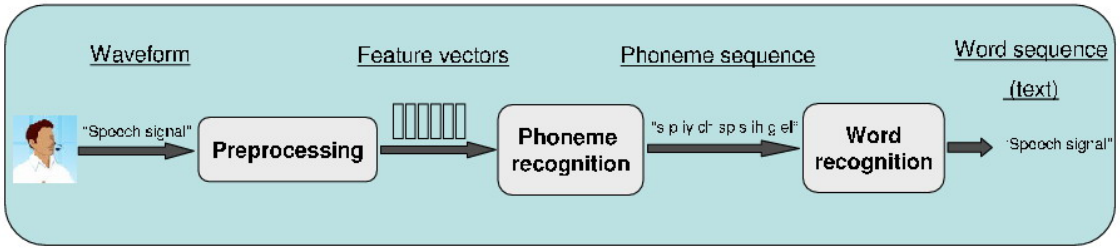


Figure 5.1: Bottom-up approach to decode the uttered waveform ***speech signal*** into its associated phoneme sequence, ***s p iy ch sp s ih g n el*** , and word sequence, ***speech signal***.

The *preprocessing* unit shown in Fig. 5.1 transforms the given speech waveform into the sequence of feature vectors at intervals of around 20 milliseconds (ms) referred to as frames. Typical speech features are mel-frequency cepstral coefficients (MFCCs) for each 20 ms of speech [51], [57], [58]. The procedure of extracting the MFCC features is provided in appendix *D*. Phoneme recognition is a crucial step in speech recognition since it forms the basis of mapping phonemes into words. Thus, it plays an important role in constructing a powerful speech recognition system.

The phonemes as the abstract underlying forms of sound are the smallest meaningful distinguishable units in a language's phonology. Each language encompasses a restricted number of phonemes [57]. Since the total number of phonemes for each language is finite, the goal of phoneme recognition is to classify a speech signal into phonemes with a given set of speech features [57], [58]. Phonemes are divided into different groups. One common method to classify phonemes is based on the way that they are produced. The classification of phonemes based on the standard TIMIT corpus is given in appendix *E* [50].

To deal with the problem of phoneme recognition, different methods such as hidden Markov models, Gaussian mixture models, and artificial neural networks have been proposed [54], [57], [58], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76], [77]. One of the proposed techniques for phoneme recognition is based on a modular system [6], [53].

To decompose the recognition task, a new concept called phoneme family was defined [6], [53]. A phoneme family, which corresponds to a module, consists of a set of similar phonemes in terms of speech features. The similarity measure of the phonemes here is defined in terms of the Euclidean distance. To obtain phoneme families, there is a need for a clustering technique. Subsequently, the performance of the selected clustering algorithm has a significant influence on the architecture and accuracy of the modular classifier. An introduction to modular approaches is provided next.

5.3 Modular Approaches

In modular approaches, a task or problem is decomposed into a number of subtasks and each module handles a subtask of the global task [78], [79], [80]. There are different motivations for using modular approaches which are as follows [81]:

- **To improve performance:** Depending on the current circumstances, modular system can switch to the most appropriate module.
- **To reduce model complexity:** By employing modular approaches, the overall system is made easier to understand, modify and extend.
- **To make the problem simple:** Sometimes the best way to solve a problem is to break it down into subtasks.
- **To recombine sensory information:** In some cases, input information comes from various independent sources or sensors. In such conditions, each module is specialized in a specific part of the input-output space.

Auda and Kamel [78] have investigated the motivations behind the modular neural networks from another point-of-view. They have reported the following four motivations:

- **Biological motivations:** Artificial neural network was firstly introduced based on the biological neurons. Artificial neural network tries to emulate the functionality of the human brain to construct useful computational approach. To built modular neural networks, several ideas have been extracted from biological systems such as modularity and cooperation among modules.
- **Psychological motivations:** Some aspects of human learning-system such as learning in stages and mixing supervised and unsupervised learning motivated modularization of the modular approaches' learning.
- **Hardware motivations:** Need to develop new architectures which have less memory and speed requirements has also motivated creating modular structures.

- **Computational motivations:** Computational complexity of an artificial neural network in terms of required training time is generally high. Modular approaches have been employed to deal with this problem.

One of the issues in modular approaches is how to divide task into several components. The task decomposition can be accomplished explicitly, automatically or by means of class decomposition [80]. Explicit decomposition is on the basis of a strong understanding of the problem and division into subtasks is known prior to training. Class decomposition is to divide k-class classification problem into 2-class classification problems. Automatic classification is carried out by the use of data partitioning techniques. Mixture of experts [82] and hierarchical mixture of experts [83] are two main approaches for task decomposition which partition the data into regions [81]. In these approaches there exists a mixture of local experts, each of which is specialized in a specific part of the input-output space during the training phase. Generally, the local experts are classifiers such as neural networks trained on a subset of the training data set [84], [85], [86], [87]. Cooperation between local experts is handled using a gating network. The architecture of these approaches are given in Figures 5.2 and 5.3.

After reviewing task decomposition techniques, we provide the methods of combining modular components. There are four different combination methods as given bellow:

1. **Cooperative combination:** In this approach, all components make some contribution to the decision.

2. **Competitive combination:** This approach selects the most appropriate module on the basis of particular circumstances corresponding to either the input or output of the modules.

   There are two main schemes for accomplishing the selection. In the first scheme, a gating network is used to output a set of scalar coefficients that serve to weight the contributions of the various inputs. These coefficients vary as a function of the input [82], [83]. The other is based on a switching scheme [81]. The switching is generally carried out based on the input.

3. **Sequential combination:** In sequential combination, the processing is successive and the computation of one module depends on the output of a preceding module.

4. **Supervisory relationship:** In this scheme, a module is used to supervise the performance of another module.

Here, the second scheme of competitive combination method is considered to construct the phoneme recognizer. To carry out the switching task, a classifier selector is used. The classifier selector, chooses a most relevant local expert (module).

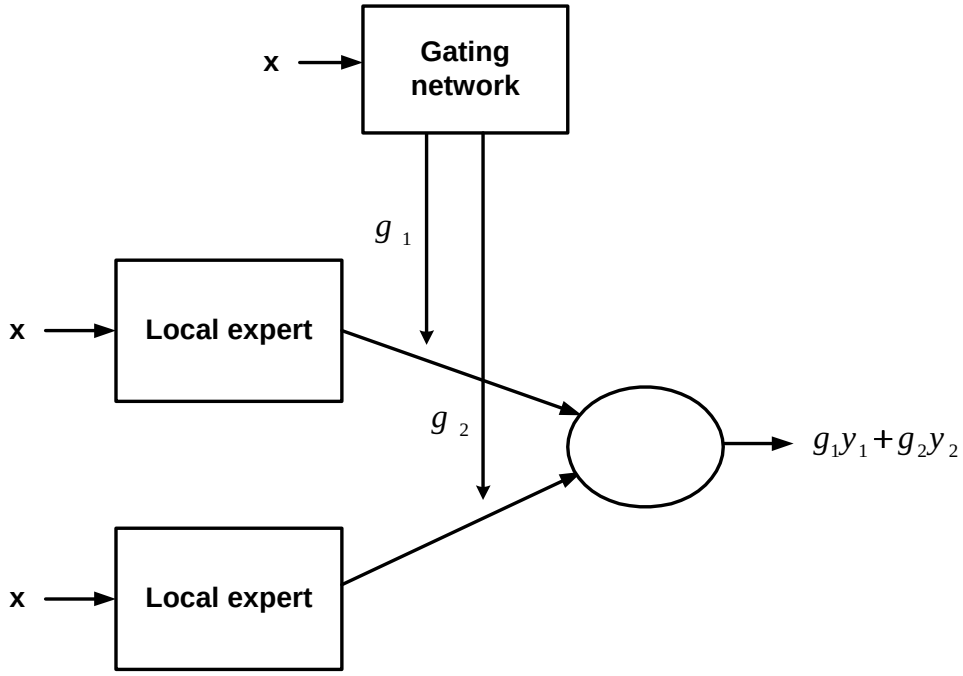


Figure 5.2: Mixture of experts

Each local expert is basically a classifier that can recognize the patterns belonging to its area of expertise.

Gaussian mixture model (GMM) is used as classifier selector and local experts. An introduction to GMM is given before providing the core approach.

5.4 Gaussian Mixture Model

Gaussian mixture model is basically considered as a density model composed of a number of components known as mixtures. The mixtures are combined to generate a multi-modal distribution by using a weighted average of multiple Gaussians.

Let's assume that each observation sequence O is defined as

$$O = (o_1, \cdots, o_n), \tag{5.1}$$

where o_t is the frame vector observed at time t. Thus, the phoneme recognition problem can be regarded as that of computing the most probable phoneme ph_k

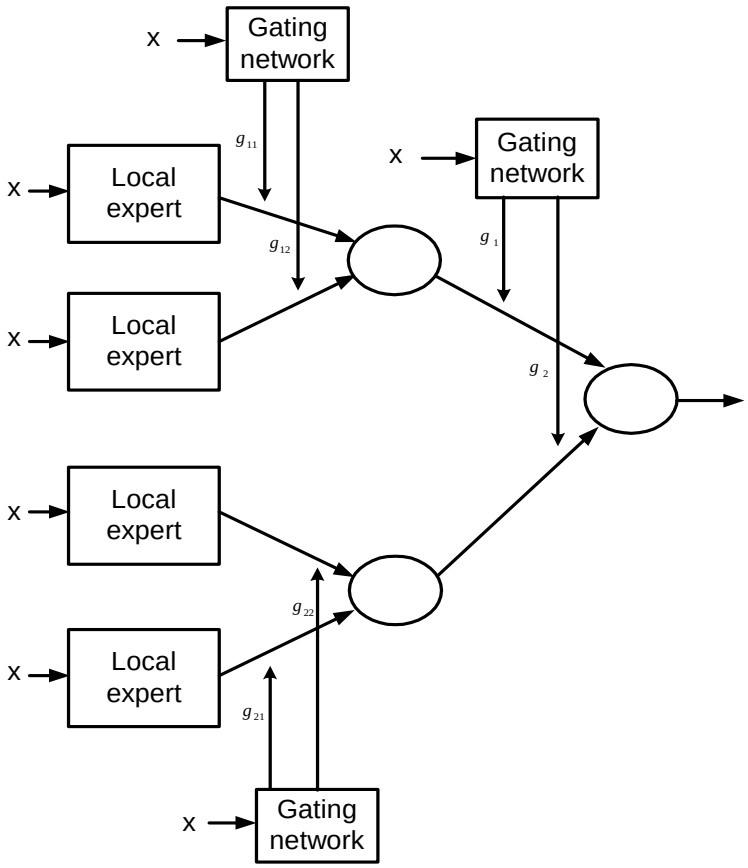


Figure 5.3: Hierarchical mixture of experts

given the observation sequence O as

$$arg\max_k\{\ P(ph_k|O)\} \tag{5.2}$$

According to Bayes' rule, $P(ph_k|O)$ can be rewritten as

$$P(ph_k|O) = \frac{P(O|ph_k)P(ph_k)}{P(O)} \tag{5.3}$$

where $P(ph_k)$ is referred to as the prior probability of phoneme ph_k. It can be easily obtained by determining its frequency in training data. $P(O)$, which denotes the probability of the observation sequence, is the same for all phonemes. Therefore, $P(ph_k|O)$ depends only on $P(O|ph_k)$. Since there is a model for each phoneme,

$P(O|ph_k)$ is estimated separately for all phonemes using corresponding models, i.e., $P(O|GMM_k);\ k = 1, \cdots, k_{ph}$, where k_{ph} denotes the number of phonemes. In other words, we have:

$$P(O|ph_k) = P(O|GMM_k),\ k = 1, \cdots, k_{ph}. \tag{5.4}$$

Moreover, $P(O|GMM_k)$ is given by

$$P(O|GMM_k) = \sum_{g=1}^{G} \pi_g \cdot \mathcal{N}(O|\mu_g, \Sigma_g), \tag{5.5}$$

where π_g indicates the weight of the g^{th} Gaussian component $\mathcal{N}(O|\mu_g, \Sigma_g)$, or simply $\mathcal{N}_g$, and

$$\sum_{g=1}^{G} \pi_g = 1,\ 0 \leq \pi_g \leq 1. \tag{5.6}$$

It is usual to consider the weighting parameter as prior probabilities. Thus, equation (5.5) can be rewritten as

$$P(O|GMM_k) = \sum_{g=1}^{G} P(g) \cdot \mathcal{N}_g. \tag{5.7}$$

In addition, Gaussian component $\mathcal{N}_g$ is obtained by

$$\mathcal{N}_g = \frac{1}{(2\pi)^{d/2}|\Sigma_g|^{1/2}} e^{-1/2(O-\mu_g)^T \Sigma_g^{-1}(O-\mu_g)}, \tag{5.8}$$

where μ_g and Σ_g denote the mean and covariance of g^{th} Gaussian component, and d is the dimensionality of data.

The associated parameters of GMM, i.e., $\pi_g,\ \mu_g$, and Σ_g, are estimated using the expectation-maximization (EM) algorithm [88], [89], [90], that aims at maximizing the likelihood of the given training set generated by the estimated probability distribution function. The likelihood function L for phoneme k is defined as

$$L_k = \prod_{i=1}^{N_{train}} P(O_i|GMM_k), \tag{5.9}$$

or equivalently its log likelihood is given by

$$ln(L_k) = \sum_{i=1}^{N_{train}} ln(P(O_i|GMM_k)), \tag{5.10}$$

where $\{O_i\}_{i=1}^{N_{train}}$ is a set of training data.

Now, the expectation and maximization steps can be specified.

- *Expectation step*: Considering initial values for the parameters of the mixture model, *partial membership* of each data point in each mixture is obtained by calculating expectation values for the membership variables of each data point. In other words, for each data point O_i and distribution $\mathcal{N}(O|g)$, the membership value $\Omega_{i,g}$ is given by

$$\Omega_{i,g}(t) = P(\mathcal{N}_g|O_i) = \frac{\pi_g P(O_i|\mathcal{N}_g, k)}{P(O_i)} = \frac{\pi_g P(O_i|\mathcal{N}_g, k)}{\sum_{j=1}^{G} \pi_j P(O_i|\mathcal{N}_j, k)}. \tag{5.11}$$

- *Maximization step*: By differentiating the log likelihood introduced in (5.10), the updating rules for GMM parameters are obtained:

$$\pi_g(t+1) = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \Omega_{i,g}(t), \tag{5.12}$$

$$\mu_g(t+1) = \frac{1}{N_{train} \cdot \pi_i(t)} \sum_{i=1}^{N_{train}} \Omega_{i,g}(t) O_i, \tag{5.13}$$

$$\Sigma_g(t+1) = \frac{1}{N_{train} \cdot \pi_i(t)} \sum_{i=1}^{N_{train}} \Omega_{i,g}(t)(O_i - \mu_g(t))(O_i - \mu_g(t))^T. \tag{5.14}$$

These steps are repeated iteratively until the termination criterion is reached.

During the recognition phase, the probability of an unknown phoneme O is evaluated given all GMMs, and the one with the maximum posterior probability is recognized as the winning phoneme label. That is,

$$k^* = arg \max_k \{P(O|GMM_k)P(ph_k)|k = 1, \cdots, k_{ph}\}. \tag{5.15}$$

The application of the multiple cooperative swarms clustering for phoneme recognition problem is next described.

5.5 Multiple Swarms Clustering for Task Decomposition

As mentioned earlier, a modular system is one of the approaches to deal with a phoneme recognition task.

In this approach, decomposing the given task into several subtasks plays an important role. The goal of decomposition is to cluster a set of phonemes into several groups called *phoneme families* which inherit similar attributes in terms of speech

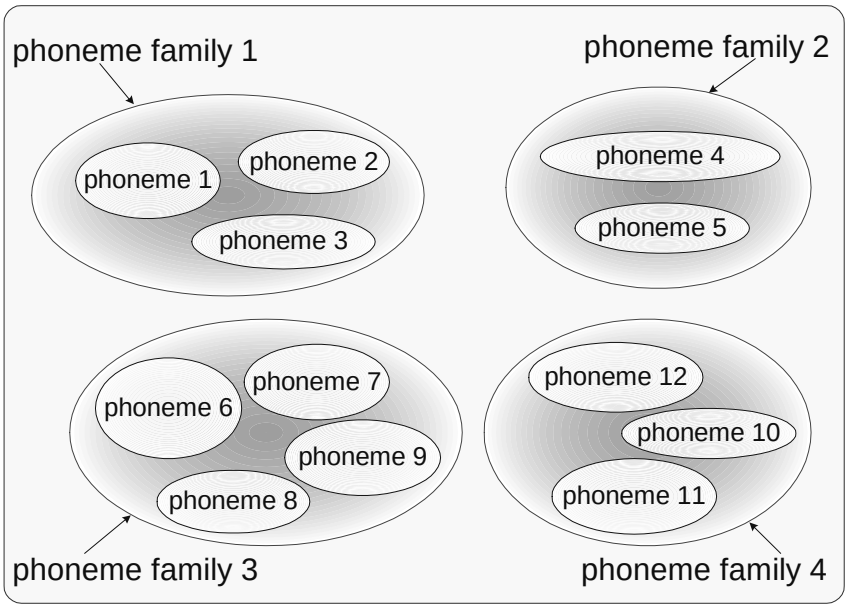


Figure 5.4: Decomposing a set of phonemes into several phoneme families

features. The concept of decomposing a set of phonemes into various phoneme families is illustrated in Fig. 5.4. We propose to use the multiple cooperative swarms clustering for dividing phonemes into different families. After applying the proposed clustering approach and determining the optimal architecture of the system, the recognition is done as follows.

An unknown phoneme is first supplied into a classifier selector that chooses the corresponding phoneme family (module) to which the given phoneme belongs. Next, the exact label of the phoneme is determined within the selected module using the associated local expert. Local expert is a classifier that is trained to recognize a certain family's phonemes. Here, Gaussian mixture models are considered as classifier selector and local experts.

A typical structure of the modular-based classifier to recognize phonemes is illustrated in Fig. 5.5.

In using the multiple cooperative swarms clustering for task decomposition, the aim is to obtain separated and compact clusters. This will enhance the recognition capability of the classifier selector as well as the overall system. The optimal decomposition of the phonemes is a structure by which the maximum accuracy for the overall system is achieved. The algorithm of the proposed approach for task decomposition is provided in Algorithm **5.1**.

After decomposing the task and obtaining the associated architecture for the

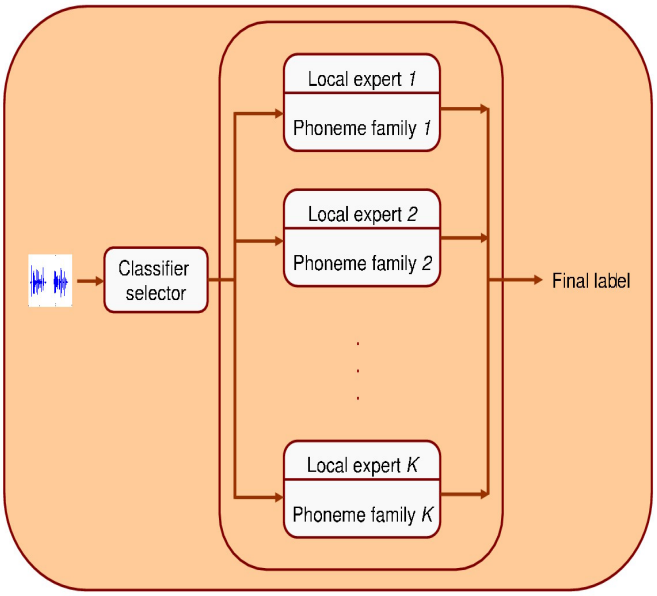


Figure 5.5: Architecture of the modular-based classifier for phoneme recognition task

Algorithm 5.1 Task decomposition in modular classifier using multiple swarms clustering

1: Provide a set of training data for each phoneme
2: **for** $b = 2$ to B **do**
3: Obtain $\tilde{C}_b = \{\tilde{C}_b^{(k)} | k = 1, 2, ..., b; \}$ by using Algorithm **3.3** (multiple cooperative swarms clustering) on the given data.
4: Train classifier selector and local experts.
5: Obtain the overall error rate of the system, E^b,
6: **end for**
7: $k^* \leftarrow arg\, min_b\{E^b\}$, where k^* is the optimal number of phoneme families which leads to minimum overall classification error.
8: Return $\tilde{C}_{k^*} = \{\tilde{C}_{k^*}^{(k)} | k = 1, 2, ..., k^*; \}$ as the optimal architecture of the modular system.

phoneme recognition, the recognition procedure needs to be described in detail. An unknown input pattern $\boldsymbol{x}$ is supplied into the system. The classifier selector chooses a phoneme family $\tilde{k}$ to which $\mathbf{x}$ belongs according to Bayes' rule:

$$\tilde{k} = arg \max_k \ \{P(\mathbf{x}|M_k)P(M_k)\}, \ k = 1, 2, ..., K, \tag{5.16}$$

where, $P(\mathbf{x}|M_k)$ denotes the posterior probability of $\mathbf{x}$ given module M and $P(M_k)$ shows prior probability of module (phoneme family) k. Moreover, different probabilities $P(\mathbf{x}|M_k)$ are determined using the associated GMMs, i.e., $P(\mathbf{x}|GMM_k)$, $k \in [1, \cdots, K]$ [91], [92].

Suppose the classifier selector transfers pattern $\mathbf{x}$ to module $M_{\tilde{k}}$. Consequently,

the associated local expert recognizes it as a member j^* of phoneme family $\tilde{k}$ based on Bayes' rule:

$$j^* = arg\max_j \; P(\mathbf{x}|M_{\tilde{k},j})P(M_{\tilde{k},j}),\; j = 1, 2, ..., m_{\tilde{k}}, \tag{5.17}$$

where, $P(\mathbf{x}|M_{\tilde{k},j})$ denotes the posterior probability of phoneme j^{th} of family $\tilde{k}$, $P(M_{\tilde{k},j})$ shows its associated prior probability and $m_{\tilde{k}}$ indicates the number of the phonemes corresponding to the phoneme family $\tilde{k}$. Again, different probabilities $P(\mathbf{x}|M_{\tilde{k},j})$ are determined using the associated GMMs, i.e., $P(\mathbf{x}|GMM_{\tilde{k},j})$. The classification algorithm includes five steps, as presented in Algorithm **5.2**.

Algorithm 5.2 Classification algorithm

1: Provide a testing data $\mathbf{x}$ to the system.
2: Compute the output of the classifier selector using equation (5.16).
3: Provide testing data $\mathbf{x}$ to phoneme family (module) $\tilde{k}$.
4: Obtain output of equation (5.17) for the testing data.
5: Return the phoneme with the maximum score as the winner phoneme.

5.6 Experimental results

In this section, the used speech database is first explained briefly. The TIMIT corpus is used to provide speech data for acquiring the acoustic-phonetic knowledge, and for developing and evaluating the ASR systems. This corpus has a total of 6300 sentences spoken by 630 male and female speakers, 10 sentences per speaker, from 8 major dialect regions of the United States of America. Moreover, three associated transcription files including text, word and phoneme are available for each sentence in addition to a speech waveform.

To evaluate the performance of the proposed approach, two data sets from TIMIT database are considered. The first data set includes 800 samples from four phonemes /aa/, /ae/, /ay/, and /el/. The second data set contains all samples of twenty phonemes /ay/, /n/, /ae/, /s/, /m/, /iy/, /r/, /axr/, /dh/, /l/, /uw/, /sh/, /z/, /f/, /v/, /ng/, /w/, /g/, /hh/, and /aa/ from dialects regions one, three and five. Also, 12 mel-frequency cepstral coefficients are used as speech features.

This section is organized in two parts. First, the multiple cooperative swarms clustering approach is compared with other clustering techniques using the first data set. As the PSO procedure has a stochastic nature, the final solution may vary for different runs. Thus, the presented results indicate the average value of 30 independent runs.

The ability of multiple cooperative swarms clustering is also studied to deal with a task decomposition problem in modular approach for phoneme recognition using the second data set. Again, 30 independent runs are used to generate each

of the presented results. To complete each run, 80% of the given data is drawn randomly to learn the architecture of the modular system and to train all the associated classifiers. The remaining 20% of the data is used to test the accuracy of the classifier selector and the overall system.

5.6.1 The performance of the multiple cooperative swarms clustering

For the first data set, the parameters are set as $w = 1.2$, $c_1 = 1.49$, $c_2 = 1.49$, $n = 10$, $w_1 = 0.85$, and $\alpha = 1.9$. Fig. 5.6 shows how the multiple cooperative swarms approach converges to an optimal solution where minimizing the objective function is desired. The average of objective value (μ) and the associated standard deviation (σ) are calculated by running the algorithm 30 times. According to

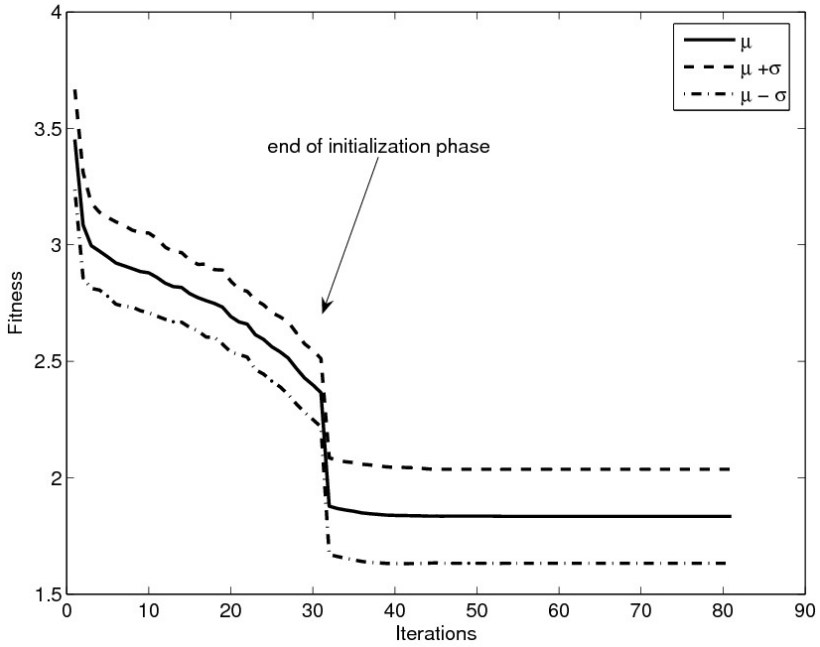


Figure 5.6: Convergence of the multiple swarms clustering approach

Fig. 5.6, a significant improvement is observed in the proposed approach in terms of combined measure right after the termination of initialization phase, where the cooperation between swarms begins.

In Fig. 5.7, the convergence of the proposed clustering approach is compared with single swarm clustering as well as K-means, K-harmonic means and fuzzy c-means clustering approaches. As shown in Fig 5.7, the convergence of both K-means and K-harmonic means clustering approaches occurs earlier than single and multiple swarms clustering approaches. However, PSO-based clustering approaches provide better results in terms of combined measure.

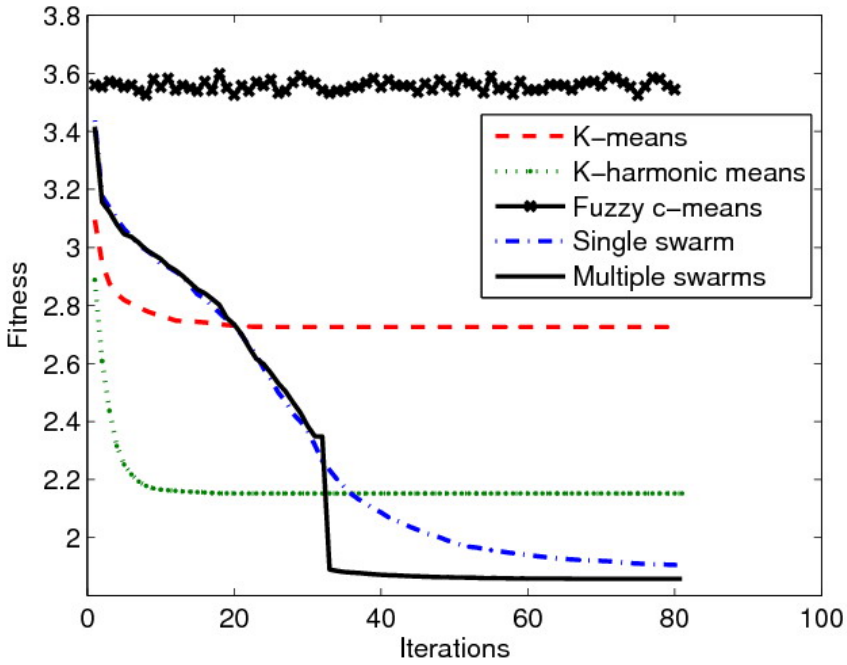


Figure 5.7: Comparing the convergence of multiple swarms clustering with other approaches in terms of combined measure

We have also compared the performance of the multiple swarms clustering with other clustering approaches in Table 5.1. This comparison is based on compactness, separation, combined measure and Turi's measures over 30 independent runs. According to Table 5.1, the proposed approach outperforms the other approaches since it uses multiple cooperating swarms and distributes the search space among multiple swarms.

We have also investigated the sensitivity of the multiple and single swarm clustering approaches to the dimensionality of feature space and the number of clusters. Fig. 5.8 presents the behavior of both single and multiple swarm clustering approaches in terms of a combined measure with regard to the dimension of feature space. As is clear from Fig. 5.8, the multiple swarms clustering approach is less

Table 5.1: Comparing proposed approach with others in terms of different validity measures

Method	Compactness	Separation	Combined measure	Turi index
K-means	**2.41e03** $\pm$ 22.6	-5.54 ± 0.2	2.75 ± 0.07	0.83 ± 0.81
KHM	$2.94e03 \pm 0.1$	-5.64 ± 0.01	2.15 ± 0.01	$3.54e05 \pm 2.62e05$
FCM	$3.56e03 \pm 0.11$	-0.46 ± 0.2	3.53 ± 0.07	$0.14e4 \pm 0.7e4$
Single PSO	$3.6e03 \pm 277.7$	$\mathbf{-18.17} \pm 0.6$	1.9 ± 0.11	-1.45 ± 0.87
Proposed	**2.41e03** $\pm$ 41.3	-18.1 ± 0.51	**1.75** $\pm$ 0.06	$\mathbf{-1.63} \pm 1.06$

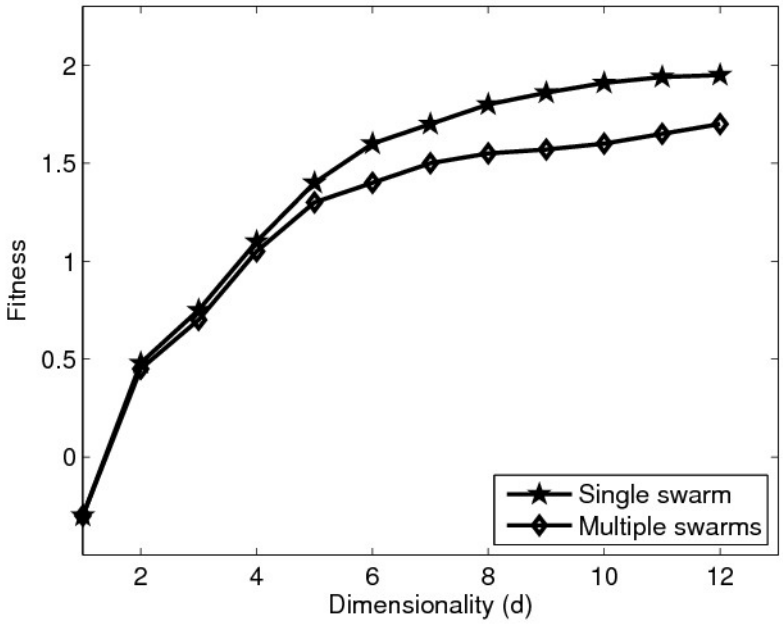


Figure 5.8: The behavior of multiple swarms and single swarm clustering approaches in terms of combined measure with regard to the dimension of feature space

sensitive to the dimensionality of feature space in comparison with a single swarm clustering approach. Similarly, the behavior of these approaches in terms of the number of clusters is shown in Fig. 5.9. Again, the results provided in Fig. 5.9 indicate that multiple swarms clustering is able to find better solutions than single swarm clustering, in terms of combined measure, as the number of clusters increases.

5.6.2 The multiple cooperative swarms for task decomposition

We have evaluated the accuracy of the classifier selector and the overall system using second data set. The changes of the accuracy in the classifier selector and the overall system are illustrated in Fig. 5.10 and Fig. 5.11 with regard to the number of clusters (or phoneme families) for combined measure and Turi's index, respectively.

Moreover, the optimal number of the clusters and the associated accuracy of classifier selector and overall system are provided in Table 5.2 for both combined measure and Turi's validity index.

As presented in Table 5.2, as compared to single swarm clustering and K-means algorithm, the multiple cooperative swarms clustering approach provides better task decomposition which leads to higher accuracy for both classifier selector and the overall system using combined measure and Turi's index.

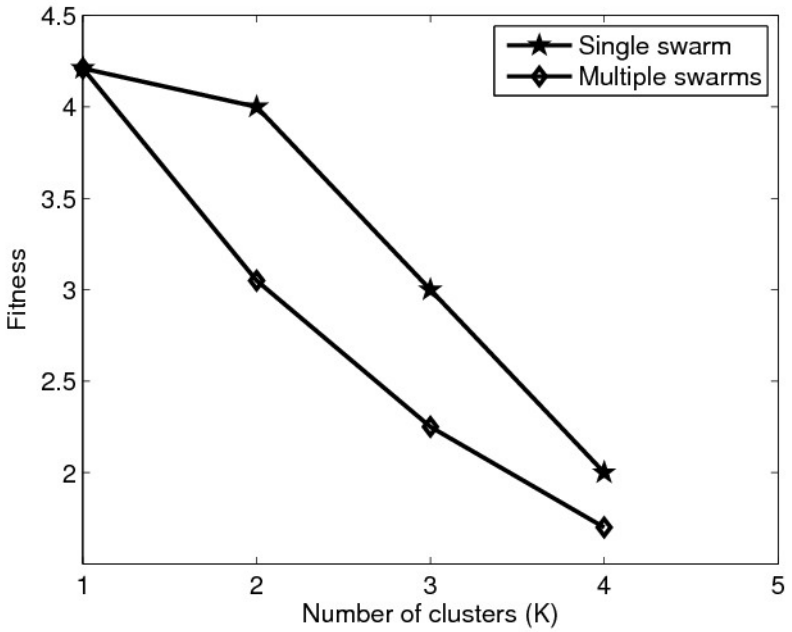


Figure 5.9: The behavior of multiple and single swarm clustering approaches in terms of combined measure with regard to the number of clusters

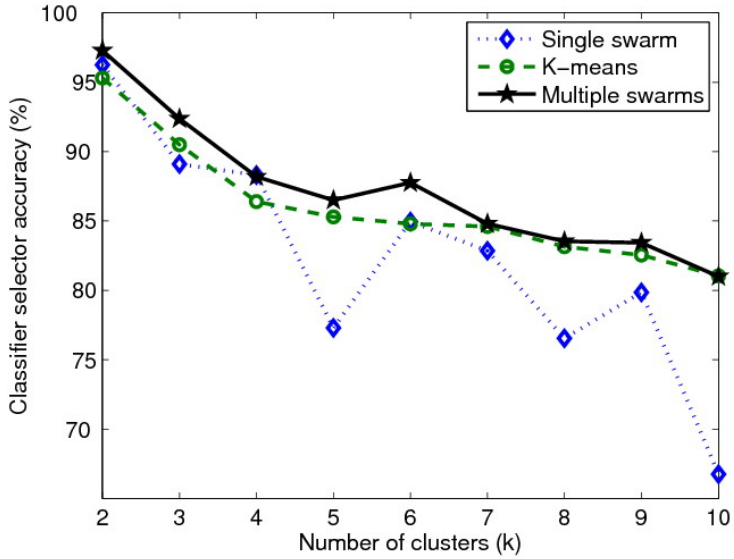
5.7 Summary

One of the critical factors for successful design of modular-based classifier for phoneme recognition is the clustering approach being used. In this chapter, it was proposed to use a multiple cooperative swarms clustering approach to decompose the phoneme recognition task into several subtasks. Each subtask coupled with a module is composed of a set of similar phonemes referred to as the phoneme family. Applying multiple cooperative swarms clustering on the TIMIT corpus indicated that the proposed approach outperforms other clustering approaches.

Table 5.2: The optimal number of the clusters and the associated accuracy of classifier selector and overall system for both combined measure and Turi's validity index

Measure		K-means	Single swarm	Multiple swarms
Combined measure	k^*	10	5	4
	Classifier selector	81.06%	77.30%	**88.30**%
	Overall system	54.85%	54.45%	**63.47**%
Turi's validity index	k^*	10	3	2
	Classifier selector	81.06%	87.80%	**97.25**%
	Overall system	54.85%	55.25%	**62.46**%

(a) Classifier selector- Combined measure



(b) Overall system- Combined measure

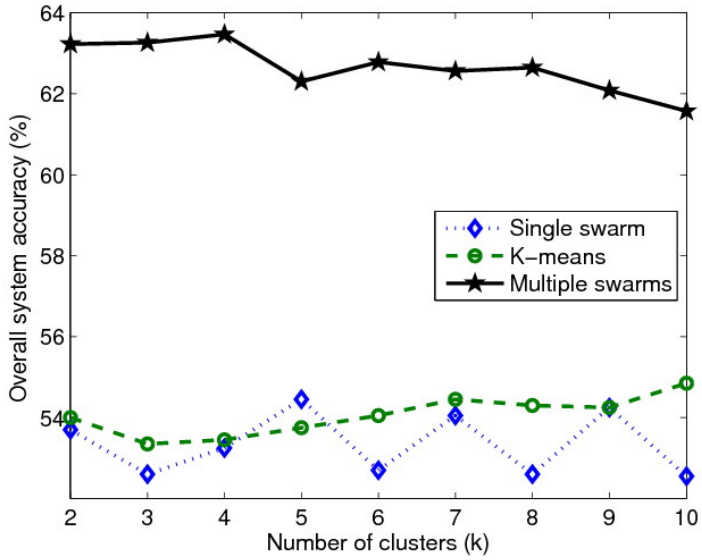
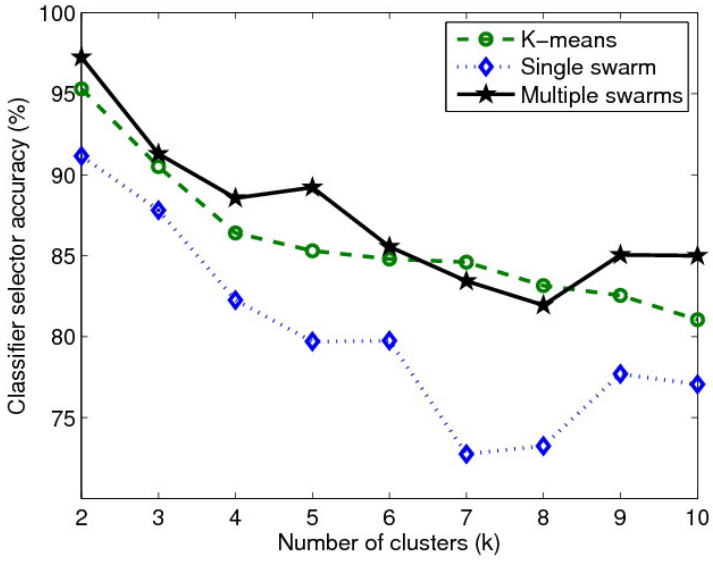


Figure 5.10: The changes of the accuracy in the classifier selector and the overall system with regard to the number of clusters using combined measure

(c) Classifier selector- Turi index



(d) Overall system- Turi index

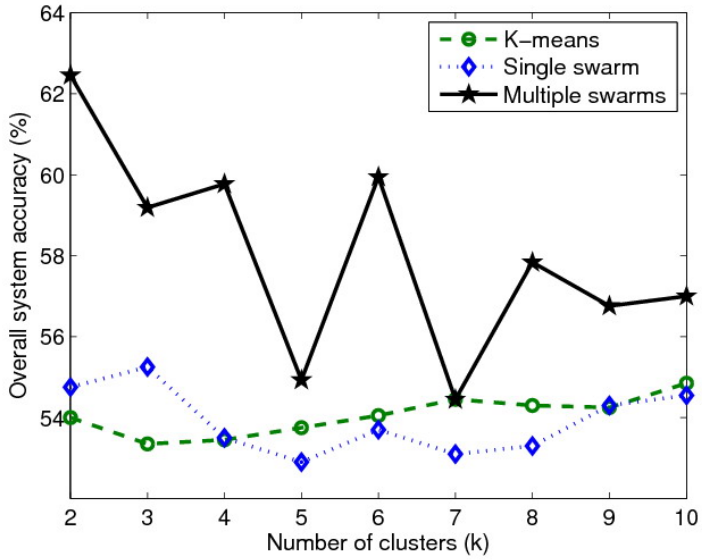


Figure 5.11: The changes of the accuracy in the classifier selector and the overall system with regard to the number of clusters using Turi's index

Chapter 6

Conclusions and Future Research Directions

In this thesis, a new clustering approach by means of multiple cooperative swarms was introduced. The proposed approach belongs to the class of PSO-based clustering techniques. PSO-based clustering techniques, in contrast to most of the partitional clustering approaches, possess the following advantages [6]:

- do not depend on the initial solutions,
- accomplish the global exploration of the search space,
- handle multiple objective functions concurrently.

Therefore, many researchers of different backgrounds have employed particle swarm optimization to resolve different types of data clustering [2], [3], [4], [8], [9], [55]. Most of the PSO-based clustering approaches were developed using a single swarm. Single swarm clustering is suitable for low dimensional data and a small number of clusters. For high-dimensional data and where the number of clusters is relatively high, it is difficult to find an optimal solution using the single swarm approach. In these situations, the multiple cooperative swarms can lead to better solutions in terms of the defined fitness function [6].

In this chapter, a summary of findings are first presented. Future research directions are next outlined. Finally, a list of publications resulting from thesis work are provided.

6.1 Summary and Conclusions

The contributions of this thesis include providing theoretical formulation of a multiple cooperative swarms approach for data clustering, detailed algorithm, stability analysis of the model order selection and application to phoneme recognition.

Multiple cooperative swarms clustering

A new technique is introduced to solve the data clustering problem based on multiple cooperative swarms. The proposed technique contains two major phases: initialization and exploitation. In the initialization phase, search space is distributed among several swarms using another swarm called a super-swarm. In other words, a part of the search space is assigned to each swarm. In the exploitation phase, each swarm searches for its corresponding cluster's center, while cooperating with other swarms. The proposed approach was applied to clustering different data sets. The proposed multiple cooperative swarm clustering outperforms the other methods because of distributing the search space among multiple swarms and using multiple cooperating swarms.

Stability-based model order selection

Similar to most of the partitional clustering approaches, multiple cooperative swarms clustering needs to know the number of clusters a priori. To enable the proposed approach to estimate the number of clusters, the stability-based approach was considered. Because the probability of providing an optimal solution by multiple cooperative swarms clustering is bigger than that of a single swarm scenario, it can provide more stable solutions than single swarm clustering. Therefore, it can provide better estimations for the model order of the underlying data. To evaluate the scalability of the proposed approach, its performance has been assessed using eight different data sets.

Application to phoneme recognition

The proposed multiple cooperative swarms clustering was successfully applied to modular classifier-based phoneme recognition. In a modular approach for phoneme recognition, decomposing the whole task into several modules plays an important role. Multiple cooperative swarms clustering yields a strong tool for decomposing the phoneme recognition task into several subtasks in modular classifier. The performance of the proposed approach was studied using the standard TIMIT corpus and it was shown that the proposed approach can produce better results.

6.2 Future Research Directions

Particle swarm optimization as a branch of swarm intelligence is in its early years and a vast number of researchers are working on different aspects of it. Accordingly, the particle swarm clustering field of research is young as well. Therefore, there is a vast number of opportunities to either improve the existing clustering algorithms or to develop new algorithms to deal with different aspects of data clustering. There are two major categories of opportunities from two points of view. From a data clustering perspective, there are different topics that need refinement and improvement. From a PSO point-of-view, there are also some issues that need to be addressed. In the following, potential research problems are explained.

- **Distributed data clustering**

  Distributed data clustering is a strong tool to deal with large-scale data sets which either are of high dimension or have huge amount of patterns. In this regard, two main strategies can be taken into consideration. One strategy is to distribute the feature space between several swarms and let the swarms collaborate and cooperate together to find the final solution. In other words, each swarm deals with a subset of feature space and the final solution is the aggregation of different swarms' solutions. The other strategy is to divide the data set into a various subsets each of which is associated with a swarm. First, each swarm finds a solution for its related subset. The final solution is then obtained by combining different solutions. To facilitate the combination of the different solutions, multiple clustering approaches can be considered.

- **Fuzzy swarm regions**

  The proposed approach assumes that the boundaries of the swarm regions remain constant during the exploitation phase which may restrict the exploration ability of the proposed approach. Considering fuzzy boundaries can resolve this problem.

- **Different notions of similarity measures** This thesis uses the Euclidean distance as a similarity measure between the patterns. However, several similarity measures are available. A comprehensive study on the effect of the different similarity measures is required to propose the best measure for different situations.

- **The selection of an appropriate cluster validity measure**

  One of the issues that affects the performance of the PSO-based clustering approaches is the fitness function being optimized. Usually, cluster validity measures are used as a fitness function. Therefore, the selection of the appropriate cluster validity measure as a fitness function is an important problem, which needs to be sought carefully.

6.3 List of Publications

This thesis has led to several publications, posters and invited talks. In the following, a list of publications is provided.

Book chapters and journal papers

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Task Decomposition Using Multiple Cooperative Particle Swarms for Phoneme Recognition*, submitted as a book chapter to Applied Swarm Intelligence, edited by Andries Engelbrecht and Martin Middendorf.

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *A Clustering Approach by Means of Multiple Cooperating Swarms*, IEEE Transactions on Evolutionary Computation, under revision.

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Model Order Selection for Data Clustering Using Multiple Cooperative Swarms*, submitted to Machine Learning Journal: Special Issue on Swarm Intelligence for Knowledge Discovery in Data.

Conference papers

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Particle Swarm Clustering Ensemble*, ACM Genetic and Evolutionary Computation Conference (GECCO-2008), 2008, Atlanta, Georgia, USA, [93].

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Model Order Selection for Multiple Cooperative Swarms Clustering Using Stability Analysis*, 2008 IEEE Congress on Evolutionary Computation (IEEE CEC 2008) within 2008 IEEE World Congress on Computational Intelligence (WCCI 2008), pp. 3387-3394, Hong Kong, [48].

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Particle Swarm-Based Approaches for Clustering Phoneme Data*, In the Proceeding of UW and IEEE Kitchener-Waterloo Section Joint Workshop, pp. 40-42, 2007, Waterloo, Canada, [56].

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Cooperative Swarms for Clustering Phoneme Data*, In the proceeding of IEEE Workshop on Statistical Signal Processing, pp. 606-610, 2007, Madison, Wisconsin, USA, [55].

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Multiple Cooperating Swarms for Data Clustering*, In the Proceeding of IEEE Swarm Intelligence Symposium, pp. 206-212, 2007, Honolulu, Hawaii, USA, [6].

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Hybrid Learning Scheme for Modular-Based Phoneme Recognizer*, IEEE International Symposium on Signal Processing and its Applications, 2007, Dubai, UAE, [53].

- Abbas Ahmadi, Fakhri Karray, Mohamed S. Kamel, *Modular-Based Classifier for Phoneme Recognition*, In the Proceeding of IEEE International Symposium on Signal Processing and Information Technology, 2006, pp. 583-588, Vancouver, British Colombia, Canada, [52].

References

[1] A.K. Jain; M.N. Murty and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, September 1999. 1, 5, 6, 7, 9, 15

[2] M. Omran ; A. Salman and A.P. Engelbrecht. Dynamic clustering using particle swarm optimization with application in image segmentation. *Pattern Analysis and Applications*, 6:332–344, 2006. 1, 2, 7, 13, 16, 20, 21, 22, 52, 77, 94

[3] M. Omran; A.P. Engelbrecht and A. Salman. Particle swarm optimization method for image clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(3):297–321, 2005. 1, 2, 13, 20, 22, 52, 77, 94

[4] X. Cui; T.E. Potok and P. Palathingal. Document clustering using particle swarm optimization. In *IEEE Proceedings of Swarm Intelligence Symposium*, pages 185–191, 2005. 1, 2, 13, 14, 20, 77, 94

[5] X. Xiao; E.R. Dow; R. Eberhart; Z.B. Miled and R.J. Oppelt. A hybrid self-organizing maps and particle swarm optimization approach. *Concurrency and Computation: Practice and Experience*, 16(9):895–915, August 2004. 1, 13, 20, 77

[6] A. Ahmadi; F. Karray and M. Kamel. Multiple cooperating swarms for data clustering. In *IEEE Swarm Intelligence Symposium*, pages 206–212, 2007. 1, 17, 52, 77, 78, 79, 94, 98

[7] A.P. Engelbrecht. *Computational Intelligence: An Introduction*. John Wiley and Sons, 2002. 2

[8] D.W. van der Merwe and A.P. Engelbrecht. Data clustering using particle swarm optimization. In *Proceeding of IEEE Congress on Evolutionary Computation*, volume 1, pages 215–220, December 2003. 2, 13, 20, 38, 52, 77, 94

[9] X. Cui; J. Gao and T. E. Potok. A flocking based algorithm for document clustering analysis. *Journal of Systems Architecture*, 52(8-9):505–515, August-September 2006. 2, 13, 52, 77, 94

[10] R.O. Duda; P.E. Hart and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2000. 5, 16, 17

[11] B. Zhang and M. Hsu. K-harmonic means: A data clustering algorithm. Technical report, Hewlett-Packard Labs, HPL-1999-124. 9, 52

[12] J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algoritms*. Plenum Press, New York, 1981. 10, 52

[13] M. Kazemian; Y. Ramezani; C. Lucas and B. Moshiri. *Swarm Intelligence in Data Mining*, chapter Swarm Clustering Based on Flowers Pollination by Artificial Bees, pages 191–202. Springer, 2006. 10

[14] J.L. Deneubourg; S. Goss; N. Franks; A. Sendova-Franks; C. Detrain and D. Chretien. The dynamics of collective sorting: robot-like ant and ant-like robot. In J.A. Meyer and S.W. Wilson, editors, *First Conference on Simulation of Adaptive Behavior*, pages 356–365. MIT Press, 1991. 11

[15] E. Lumber and B. Faieta. Density and adaptation in populations of clustering ants. In *Third European Conference on Simulation of Adaptive Behavior*, pages 499–508, 1994. 11

[16] N. Monmarche; M. Silmane and G. Venturini. On improving clustering in numerical databases with artificial ants. In *5th European Conference on Advances on Artificial Life*, volume 1674, pages 626–635, 1999. 11

[17] M. Kanade and L.O. Hall. Fuzzy ants as a clustering concept. In *22th International Conference of the North American Information Processing Society*, pages 227–235, 2003. 11

[18] A.P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley and Sons, 2005. 11, 12, 13, 17, 20, 22

[19] J. Kennedy; R.C. Eberhart and Y. Shi. *Swarm Intelligence*. Morgan Kaufman Publishers, 2001. 11, 20, 21

[20] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. 11, 20, 52

[21] U. Paquet. Training support vector machines with particle swarms. Master's thesis, Department of Computer Science of Pretoria, 2003. 12

[22] A. Abraham; H. Guo and H. Liu. *Studies in Computational Intelligence*, chapter Swarm Intelligence: Foundations, Perspectives and Applications. Springer-Verlag, Germany, 2006. 13, 40

[23] X. Xiao; E.R. Dow; R. Eberhart; Z.B. Miled and R.J. Oppelt. Gene clustering using self-organizing maps and particle swarm optimization. In *IEEE Proceedings of International Parallel Processing Symposium*, page 10pp, 2003. 13, 20, 52

[24] N. Holden and A.A. Freitas. A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data. In *IEEE Swarm Intelligence Symposium*, pages 100–107, 2005. 13

[25] M. Halkidi; Y. Batistakis and M. Vazirgiannis. On clustering validation techniques. *Intelligent Information Systems*, 17(2-3):107–145, 2001. 16, 18

[26] R.H. Turi. *Clustering-Based Colour Image Segmentation*. PhD thesis, Monash University, Australia, 2001. 17

[27] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Cybernetics*, 3:32–57, 1973. 18

[28] M. Halkidi and M. Vazirgiannis. Clustering validity assessment: finding the optimal partitioning of a data set. In *International Conference on Data Mining*, pages 187–194, 2001. 18

[29] R.C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *6th International Symposium on Micro Machine and Human Science*, pages 39–43, Nagoya, Japan, 1995. 20, 77

[30] F.O. Karray and C.W. De Silva. *Soft Computing and Intelligent Systems Design: Theory, Tools and Applications*. Addison-Wesley, 2004. 20

[31] F. Ye and C. Chen. Alternative kpso-clustering algorithm. *Tamkang Journal of Science and Engineering*, 8(2):165–174, 2005. 20

[32] C. Chen and F. Ye. Particle swarm optimization algorithm and its application to clustering analysis. In *IEEE International Conference on Networking, Sensing and Control*, pages 789–794, 2004. 20

[33] A. Abraham; C. Grosan and V. Ramos, editors. *Swarm Intelligence in Data Mining*. Springer, 2006. 21

[34] W.M. Schaffer; D.W. Zeh; S.L. Buchmann; S. Kleinhaus; M.V. Schaffer and J. Antrim. Competition for nectar between introduced honeybees and native north american bees and ants. *Ecology*, 64:564–577, 1983. 21

[35] J. Moore and R. Chapman. Application of particle swarm to multiobjective optimization. Technical report, Department of Computer Science and Software Engineering, Auburn University, 1999. 22

[36] C.A. Coello Coello and M.S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization algorithm. In *IEEE Congress on Evolutionary Computation*, volume 2, pages 1051–1056, 2002. 22

[37] C.A. Coello Coello; G. Toscano Pulido and M.S. Lechuga. An extension of particle swarm optimization that can handle multiple objectives. In *Workshop on Multiple Objective Metaheuristics*, pages 1–5, 2002. 22

[38] K.E. Parsopoulos and M.N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *ACM Symposium on Applied Computing*, pages 603–607, 2002. 22

[39] X. Hu and R.C. Eberhart. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *IEEE Congress on Evolutionary Computation*, volume 2, pages 1666–1670, 2002. 22

[40] J.E. Fieldsend and S. Singh. A multi-objective algorithm based upon particle swarm optimization. In *UK Workshop on Computational Intelligence*, pages 37–44, 2003. 22

[41] X. Hu; R.C. Eberhart and Y. Shi. Particle swarm with extended memory for multiobjective optimization. In *IEEE Swarm Intelligence Symposium*, pages 193–197, April 2003. 22

[42] F. van den Bergh and A.P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computing*, 8(3):225–239, June 2004. 22

[43] M. El-Abd and M. Kamel. Information exchange in multiple cooperating swarms. In *Proceedings of IEEE Swarm Intelligence Symposium*, pages 138–142, 2005. 22

[44] T. Blackwell and J. Branke. *Applications of Evolutionary Computing*, chapter Multi-Swarm Optimization in Dynamic Environments, pages 488–599. Springer, 2004. 22

[45] M.J. Panik. *Advanced Statistics from an Elementary Point of View*. Elsevior Academic Press, 2005. 32, 112

[46] C.L. Blake and C.J. Merz. UCI Repository of Machine Learning Databases. *http://www.ics.uci.edu/ mlearn/MLRepository.html*, 1998. 38, 61

[47] T. Lange; V. Roth; M.L. Braun and J.M. Buhmann. Stability-based validation of clustering solutions. *Neural Computing*, 16:1299–1323, 2004. 52, 53, 54, 55, 57

[48] A. Ahmadi; F. Karray and M.S. Kamel. Model order selection for multiple cooperative swarms clustering using stability analysis. In *IEEE Congress on Evolutionary Computation within IEEE World Congress on Computational Intelligence*, pages 3387–3394, Hong Kong, 2008. 53, 54, 97

[49] L.I. Kuncheva and D.P. Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1798–1808, Nov. 2006. 53

[50] National Institute of Standards and Technology. *TIMIT Acoustic-Phonetic Continuous Speech Corpus*, Speech Disc 1-1.1, NTIS Order No. PB91-5050651996, October 1990. 61, 78

[51] M. Seltzer. Sphinx III signal processing front end specification. Technical report, CMU Speech Group, 1999. 61, 78, 114

[52] A. Ahmadi ; F. Karray and M. Kamel. Modular-based classifier for phoneme recognition. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 583–588, August 2006. 76, 98

[53] A. Ahmadi; F. Karray and M.S. Kamel. Hybrid learning scheme for modular-based phoneme recognizer. In *Proceeding of International Symposium on Signal Processing and its Applications*, pages 1–4, Dubai, UAE, 2007. 76, 78, 79, 98

[54] L. Mesbahi and A. Benyetto. Continuous speech recognition by adaptive temporal radial basis function. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 574–579, 2004. 77, 78

[55] A. Ahmadi; F. Karray and M.S. Kamel. Cooperative swarms for clustering phoneme data. In *Proceeding of IEEE Workshop on Statistical Signal Processing(SSP-2007)*, pages 606–610, 2007. 77, 94, 98

[56] A. Ahmadi; F. Karray and M.S. Kamel. Particle swarm-based approaches for clustering phoneme data. In *UW and IEEE Kitchener-Waterloo Section Joint Workshop*, pages 40–42, 2007. 77, 97

[57] L. Deng and D. O'Shaughnessy. *Speech Processing: A Dynamic and Optimization-Oriented Approach*. Marcel Dekker Inc., 2003. 78

[58] J. Deller; J. Hansen and J. Proakis. *Discrete-Time Processing of Speech Signals*. IEEE Press, 2000. 78

[59] L. Rabiner and R. Schafer. *Digital Processing of Speech Signals*. Prentic Hall, 1978. 78

[60] J. Coleman. *Introducing Speech and Language Processing*. Cambridge University Press, 2005. 78

[61] A. Waibel; T. Hanazawa; G. Hinton; K. Shikano; K. Shikano and L. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics,Speech and Signal Processing*, 37(3):328–339, 1989. 78

[62] M.R. Berthold. A time delay radial basis function for phoneme recognition. In *International Conference on Neural Networks*, pages 4470–4472, 1994. 78

[63] A. Agbago and C. Barriere. Fast two-level-dynamic-programming algorithm for speech recognition. In *ICASSP*, volume 5, pages V– 129–32, 2004. 78

[64] K. Aikawa. Speech recognition using time-warping neural networks. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 337–346, 1991. 78

[65] K. Aikawa. Time-warping neural network for phoneme recognition. In *IEEE International Joint Conference on Neural Networks*, pages 2122–2127, 1991. 78

[66] A. Waibel; T. Hanazawa; G. Hinton; K. Shikano; K. Shikano and L. Lang. Phoneme recognition: neural networks vs. hidden markov models. In *ICASSP*, pages 107–110, 1988. 78

[67] A. Glaeser. Compact modular neural networks in a hybrid speaker-independent speech recognition system. In *IEEE International Conference on Neural Networks*, pages 1895–1899, 1996. 78

[68] T.D. Harrison and F. Fallside. A connectionist model for phoneme recognition in continuous speech. In *ICASSP*, pages 417–420, 1989. 78

[69] R.S. Bajwa and R.M. Owens. Simultaneous speech segmentation and phoneme recognition using dynamic programming. In *ICASSP*, pages 3213–3216, 1996. 78

[70] A. Ganapathiraju; J.E. Hamaker and J. Picone. Applications of support vector machines to speech recognition. *IEEE Transactions on Signal Processing*, 52(8):2348–2355, 2004. 78

[71] I. Kirschning and H. Tomabechi. Phoneme recognition using a time-sliced recurrent recognizer. In *IEEE International Conference on Neural Networks*, pages 4437–4441, 1994. 78

[72] Y. Ariki; F.R. McInnes and M.A. Jack. Hierarchical phoneme recognition by hidden markov models based on multiple feature integration. *Electronics Letters*, 25(14):918–919, 1989. 78

[73] S.E. Golowich and D.X. Sun. A support vector/hidden markov model approach to phoneme recognition. In *ASA Proceedings of the Statistical Computing Section*, pages 125–130, 1998. 78

[74] J. Takami and S. Sagayama. A pairwise discriminant approach to robust phoneme recognition by time-delay neural networks. In *ICASSP*, pages 89–92, 1991. 78

[75] A.J. Robinson. An application of recurrent nets to phone probability estimation. *IEEE Transactions on Neural Networks*, 5(2):298–305, 1994. 78

[76] M. Watts and N. Kasabov. Simple evolving connectionist systems and experiments on isolated phoneme recognition. In *IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pages 3213–3216, 2000. 78

[77] T.J. Reynolds and E.B. Pizzolato. Phoneme classification with multinets. In *International Conference on Signal Processing Proceedings*, pages 706–709, 1998. 78

[78] G. Auda and M. Kamel. Modular neural networks: A survey. *International Journal of Neural Systems*, 9(2):129–151, 1999. 79

[79] G. Auda and M. Kamel. CMNN: Cooperative Modular Neural Networks. *Neurocomputing*, 20:189–207, 1998. 79

[80] B. Lu and M. Ito. Task decomposition and module combination based on class relations: A modular neural network for pattern classification. Technical report, Bio-Mimetic Control Research Center, 1998. 79, 80

[81] A.J.C. Sharkey. *Combining Artificial Neural Networks*. Springer, 1999. 79, 80

[82] R.A. Jacobs; M.I. Jordan; S.J. Nowlan and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–97, 1991. 80

[83] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994. 80

[84] G. Joydeep. Multi-classifier systems: Back and future. In *Third International Workshop on Multiple Classifier Systems*, pages 1–15, 2002. 80

[85] G. Giacinto; F. Roli and G. Vernazza. *Design of Multiple Classifier Systems*. PhD thesis, University of Salerno, 1998. 80

[86] S. Yang; A. Browne and P.D. Picton. Multistage neural network ensembles. In *Third Int'l Workshop on Multiple Classifier Systems*, pages 91–97, 2002. 80

[87] J. Kittler. *Soft Computing Approach to Pattern Recognition and Image Processing*, chapter Multi classifier systems, pages 3–22. World Scientific, 2002. 80

[88] C. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006. 83

[89] A. Dempster; N. Laird and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):138, 1997. 83

[90] F. Sha and L.K. Saul. Large margin gaussian mixture modeling for phonetic classification and recognition. In *ICASSP 2006*, volume 1, pages 265–268, 2006. 83

[91] R. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley and Sons, 1992. 86

[92] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001. 86

[93] A. Ahmadi; F. Karray and M.S. Kamel. Particle swarm clustering ensemble. In *ACM Genetic and Evolutionary Computation Conference (GECCO-2008)*, pages 159–160, Atlanta, Georgia, USA, 2008. 97

[94] Y. Li. Singal processing for speech applications. Language Technology Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, 2004. 115

Appendix A

Explaining why the probability of achieving an optimal solution decreases with increasing dimensionality

Consider the following optimization problem

$$\begin{array}{l} \mathcal{Z} = \min \ f(x) \\ \text{s.t.}: \quad x \in \mathsf{S}, \end{array} \tag{A.1}$$

where S is search space, or feasible solution region. Assume S is a d-dimensional hyper-sphere of radius R. Suppose also the optimal solution is located in a smaller d-dimensional hyper-sphere of radius r (Fig. A.1). We know that the volume of a d-dimensional hyper-sphere of radius R is obtained by

$$V(R,d) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)} R^d, \tag{A.2}$$

where $\Gamma(.)$ stands for gamma function given by

$$\Gamma(d) = \int_0^\infty x^{d-1} e^{-x} dx. \tag{A.3}$$

The probability of finding a solution in the optimal region using any search technique is as follows:

$$P_r(\text{converge to an optimal solution}) = \frac{V(r,d)}{V(R,d)}. \tag{A.4}$$

Using equation (A.3), the above-mentioned equation is simplified as

$$P_r(\text{converge to optimal solution}) = (\frac{r}{R})^d. \tag{A.5}$$

In other words, the probability of finding an optimal solution decreases by increasing the dimensionality of data provided r and R remain constant.

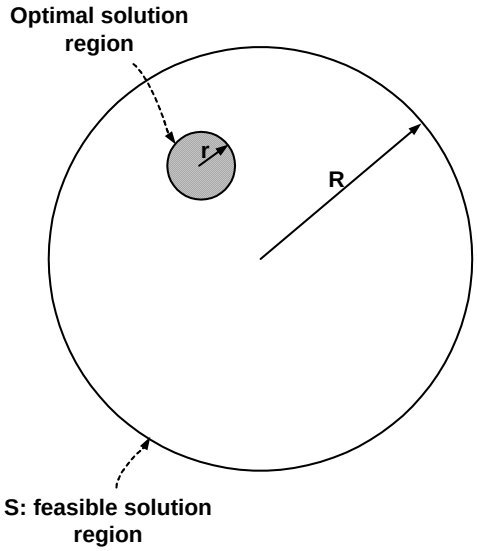


Figure A.1: Feasible and optimal solution regions

Appendix B

Proof of effectiveness of multiple swarms over single swarm in both higher dimensions and larger number of clusters

B.1 Single swarm

Assume we tend to cluster data into K different clusters, including $C^1, ..., C^K$. Let's suppose the feasible solution region is a d-dimensional hyper-sphere of radius R and the optimal solutions for centers of the clusters are located in K different d-dimensional hyper-spheres of radii $r_1, ..., r_K$, respectively. Moreover, $\mathbf{m}^1, ..., \mathbf{m}^K$ are the corresponding centers of the clusters. To achieve an optimal solution, these centers should be chosen from optimal solution regions. In the case of a single-swarm, we denote P_r(converge to optimal solution) by P_r^1 computed as follows

$$P_r^1 = \prod_{k=1}^{K} P_r(\mathbf{m}^k \in C^k), \tag{B.1}$$

where $P_r(\mathbf{m}^k \in C^k)$ stands for the probability of selecting the center of cluster k from its corresponding optimal region defined by

$$P_r(\mathbf{m}^k \in C^k) = (\frac{r}{R})^d. \tag{B.2}$$

Using this expression, equation (B.1) can be rewritten as follows:

$$P_r^1 = \prod_{k=1}^{K} (\frac{r_k}{R})^d. \tag{B.3}$$

By simplifying, the probability of converging to an optimal solution by a single swarm can be calculated by

$$P_r^1 = \frac{(r_1 r_2 \cdots r_K)^d}{R^{d.K}}. \tag{B.4}$$

B.2 Multiple swarms

In the case of multiple-swarms, each swarm explores a part of the feasible solution region characterized by a $d-$dimensional hyper-sphere of radius R_k. As $R_k < R$ for all k, the following inequality is valid:

$$R_1 \cdots R_K < R^K. \tag{B.5}$$

Since $d \geq 1$, inequality (B.5) can be modified as

$$(R_1 \cdots R_K)^d < R^{d.K}. \tag{B.6}$$

Each swarm searches its corresponding cluster's center. Similar to the single swarm case, assume the optimal solution for each swarm k is situated in a $d-$dimensional hyper-sphere of radius r_k. Accordingly, the probability of getting an optimal solution using multiple swarms at each iteration (denoted by P_r^M) is calculated as follows:

$$P_r^M = \prod_{k=1}^{K} P_r(\mathbf{m}^k \in C^k)\cdot \tag{B.7}$$

It can be simplified as

$$P_r^M = \prod_{k=1}^{K} (\frac{r_k}{R_k})^d = \frac{(r_1 r_2 \cdots r_K)^d}{(R_1.R_2 \cdots R_K)^d}. \tag{B.8}$$

According to equations (B.4) and (B.8), we have

$$\frac{P_r^M}{P_r^1} = \frac{R^{d.K}}{(R_1.R_2 \cdots R_K)^d}. \tag{B.9}$$

Considering equation (B.6), it is proved that

$$\frac{P_r^M}{P_r^1} > 1\cdot \tag{B.10}$$

In other words, $P_r^M > P_r^1\cdot$

B.3 Higher dimensions and larger number of clusters

By defining $\beta = \frac{P_r^M}{P_r^1}$, equation (B.9) can be rewritten as

$$\beta = \frac{R^{d.K}}{(R_1.R_2 \cdots R_K)^d}. \tag{B.11}$$

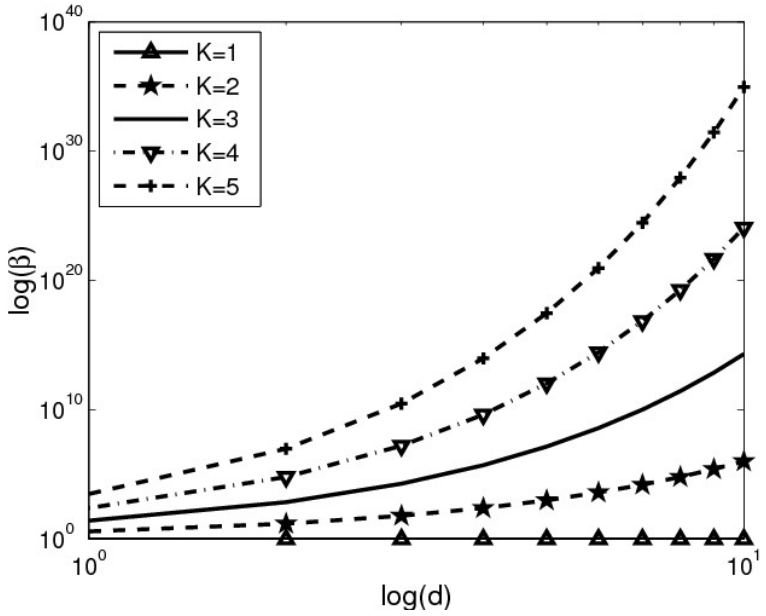


Figure B.1: Relation between β and both dimensionality (d) and number of clusters (K)

β indicates the ratio of the probability of finding an optimal solution using multiple swarms to the probability of finding an optimal solution by a single swarm. In the previous section, we proved that β is greater than one. We now examine the influences of increasing both dimensionality of data (d) and the number of clusters (K) on β. For the sake of simplicity, let's suppose

$$R_1 = R_2 = \cdots = R_K = \frac{1}{K}R\cdot \tag{B.12}$$

Considering expression (B.12), we can rewrite equation (B.11) as

$$\beta = K^{d.K}\cdot \tag{B.13}$$

In Fig. B.1, we have illustrated the relation between β and both dimensionality of data and number of clusters.

As illustrated in Fig. B.1, by increasing the dimensionality of data and the number of clusters, β increases. In other words, the ratio of probability of finding an optimal solution using multiple swarms versus a single swarm grows up by increasing the dimensionality of data and the number of clusters.

Appendix C

Evaluating the statistical significance of the obtained results

The statistical significance of the obtained results using T-test [45] is provided in Table C.1 in terms of p-value which is the probability of observing the given sample results under the assumption that the null hypothesis is true. The equality of the mean of two samples at a significance level γ is considered as the null hypothesis. The null hypothesis is rejected if the obtained p-value is less than the typical significance level of $\gamma = 5\%$. In Table C.1, D1, D2, ..., D8, correspond to speech, zoo, breast cancer, wine, glass, iris, teaching assistant evaluation, and diabetes, respectively.

Table C.1: Statistical significance of the obtained results using T-test in terms of p-value

Cluster validity measure/ Clustering method		Data sets							
		D1	D2	D3	D4	D5	D6	D7	D8
Compactness	K-means	0.5	0.487	0.48	0.001	0	0.01	0	0
	KHM	0	0.2	0	0	0	0	0	0
	FCM	0	0	0	0	0	0.3	0	0.001
	Hybrid PSO	0.0001	0.2	0	0	0.1	0.18	1e-06	0
	Single swarm	0	0.1	0.02	0.07	0	3e-06	3e-05	0
Separation	K-means	0	0	0	0	0	0	0	0
	KHM	0	0	0	0	0	0	0	0
	FCM	0	0	0	0	0	0	0	0
	Hybrid PSO	0	0	0	0	0	0	0	0
	Single swarm	0.3	0.01	0	0.1	0.3	0.03	0.09	0.2
Combined measure	K-means	0	0.03	0	0	0	0.0001	0	0
	KHM	0	0.19	0	0	0	0.006	0	0
	FCM	0	0	0	0	0	0.108	0	0
	Hybrid PSO	8e-06	0.1	0	0	0.4	7e-06	0	0
	Single swarm	0.001	0.04	0.26	0.001	0	0.03	0.1	0.006
Turi's index	K-means	0	0	0	0.003	0	0	0	0
	KHM	0	0	0	0	0	0	0	0
	FCM	0	4e-05	0	0.002	0	0	0	0
	Hybrid PSO	0	0	0	0.003	0	0	0	0
	Single swarm	0.19	0.4	0.3	8e-06	0.03	0.4	0.49	0.048

Appendix D

MFCC and delta delta MFCC features

The steps to construct MFCC features are as follows [51]:

1. Pre-Emphasis:
   The following FIR pre-emphasis filter is applied to the input waveform:
   $$y[n] = x[n] - \alpha x[n-1] \tag{D.1}$$
   α is provided by the user or set to the default value. If $\alpha = 0$, then this step is skipped. In addition, the appropriate sample of the input is stored as a history value for use during the next round of processing.

2. Windowing: The frame is multiplied by the following Hamming window:
   $$w[n] = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) \tag{D.2}$$
   N is the length of the frame.

3. Power Spectrum
   The power spectrum of the frame is computed by performing a DFT of length specified by the user, and then computing its magnitude squared.
   $$S[k] = (real(X[k]))^2 + (imag(X[k]))^2 \tag{D.3}$$

4. Mel Spectrum
   The mel spectrum of the power spectrum is computed by multiplying the power spectrum by each of the of the triangular mel weighting filters and integrating the result.
   $$\widetilde{S}[l] = \sum_{k=0}^{N/2} S[k]M_l[k] \quad l = 0, 1, ..., L-1; \tag{D.4}$$
   N is the length of the DFT, and L is total number of triangular mel weighting filters.

5. Mel Cepstrum

   A DCT is applied to the natural logarithm of the mel spectrum to obtain the mel cepstrum:

$$c[n] = \sum_{i=0}^{L-1} ln(\widetilde{S}[i]) \cos(\frac{\pi n}{2L}(2i+1)) \quad n = 0, 1, ..., C-1; \tag{D.5}$$

   C is the number of cepstral coefficients.

Delta MFCC
Also, delta MFCC coefficients can be calculated using the following equation [94]:

$$\Delta c[n] = c[n+1] - c[n] \tag{D.6}$$

Delta delta MFCC
Moreover, to obtain delta-delta coefficients, following equation will be applied [94]:

$$\Delta\Delta c[n] = \Delta c[n+1] - \Delta c[n] \tag{D.7}$$

Appendix E

Categorization of the phonemes based on TIMIT database

According to TIMIT database categorization, English phonemes are classified into following types. For each phoneme, an example is given as well

1. Closure
    - bcl, dcl, gcl, pcl, tcl, kcl
2. Stops
    - b: bee
    - d: day
    - g: geese
    - p: pea
    - t: tea
    - k: key
    - dx: muddy, dirty
    - q: at, bat
3. Fricatives
    - s: sea
    - sh: she
    - z: zone
    - zh: azure
    - f: fin
    - th: thin

- v: van
- dh: then

4. Nasals
    - m: moon
    - n: noon
    - ng: sing
    - em: bottom
    - en: button
    - eng: Washington
    - nx: winner
5. Affricates
    - jh: joke
    - ch: choke
6. Semivowels
    - l: lay
    - r: ray
    - w: way
    - y: yacht
7. Vowels
    - iy: beet
    - ih: bit
    - eh: bet
    - ey: bait
    - ae: bat
    - aa: bottom
    - aw: bout
    - ay: bite
    - ah: but
    - ao: bougth
    - oy: boy
    - ow: boat
    - uh: book

- uw: boot
- ux: toot
- el: bottle
- er: bird
- ax: about
- ix: debit
- axr: butter
- ax-h: suspect

8. Aspirations
    - hh: hay
    - hv: ahead