DETERMINING THE NUMBER OF CLUSTERS AND DISTINGUISHING OVERLAPPING CLUSTERS IN DATA ANALYSIS

By

Haojun Sun

submitted in partical fulfillment of the
requirements for the degree of
Doctor of Philosophy
at
Faculté des Sciences
Université de Sherbrooke
Sherbrooke, Québec, Canada
Oct. 2004

Le 25 avril 2005

le jury a accepté la thèse de M. Haojun Sun dans sa version finale.

Membres du jury

M. Shengrui Wang
Directeur
Département d'informatique

M. François Deschênes
Membre
Département d'informatique

M. François Dubeau
Membre
Département de mathématiques

M. Luis Rueda
Membre externe
School of Computer Science - University of Windsor

M. Bernard Colin
Président-rapporteur
Département de mathématiques

Sommaire

Le processus de Clustering permet de construire une collection d'objets (clusters) similaires au sein d'un même groupe, et dissimilaires quand ils appartiennent à des groupes différents. Dans cette thèse, on s'intéresse à deux problèmes majeurs d'analyse de données: 1) la détermination automatique du nombre de clusters dans un ensemble de données dont on a aucune information sur les structures qui le composent; 2) le phénomène de recouvrement entre les clusters.

La plupart des algorithmes de clustering souffrent du problème de la détermination du nombre de clusters qui est souvent laissé à l'utilisateur. L'approche classique pour déterminer le nombre de clusters est basée sur un processus itératif qui minimise une fonction objectif appelé indice de validité. Notre but est de: 1) développer un nouvel indice de validité pour mesurer la qualité d'une partition, qui est le résultat d'un algorithme de clustering; 2) proposer un nouvel algorithme de clustering flou pour déterminer automatiquement le nombre de clusters. Une application de notre nouvel algorithme est présentée. Elle consiste à la sélection des caractéristiques dans une base de données.

Le phénomène de recouvrement entre les clusters est un des problèmes difficile dans la reconnaissance de formes statistiques. La plupart des algorithmes de clustering ont des difficultés à distinguer les clusters qui se chevauchent. Dans cette thèse, on a développé une théorie qui caractérise le phénomène de recouvrement entre les clusters dans un modèle de mélange Gaussien d'une manière formelle. A partir de cette théorie, on a développé un nouvel algorithme qui calcule le degré de recouvrement entre les clusters dans le cas multidimensionnel. Dans ce cadre précis, on a étudié les facteurs qui affectent la valeur théorique du degré de recouvrement. On a démontré comment cette théorie peut être utilisée pour la génération des données de test valides et concrètes pour une évaluation objective des indices de validité par rapport à leurs capacités à distinguer les clusters qui se chevauchent. Finalement, notre théorie est utilisable dans une application de segmentation des images couleur en utilisant un algorithme de clustering hiérarchique.

Abstract

Data clustering is the process of grouping the data into clusters so that the objects within a cluster are highly similar and the objects in different clusters are highly dissimilar. The main focus of this thesis is to investigate into two important problems in clustering: determining the number of clusters in a given data set and studying the phenomenon of overlapping between clusters.

Determining the number of clusters is one of the most important topics in cluster analysis. A common approach for determining the number of clusters is an iterative trial-and-error process based on a cluster validity index. One of the main goal of this thesis is to develop a new validity index for measuring the "goodness" of trial-clustering and an effective fuzzy algorithm for automatically determining the number of clusters. An application of the new algorithm in subset feature selection is proposed also.

The phenomenon of cluster overlap is present in real applications. Many algorithms fail to distinguish overlapping clusters. In this thesis, we establish a theory on the overlap phenomenon in the case of the Gaussian mixture, a fundamental data distribution model for many clustering algorithms. Based on this theory, we develop an algorithm for calculating the overlap rate between two clusters and investigate factors that affect the value of the overlap rate. We show how the theory can be used to generate truthed data sets for evaluating the ability of a validity index for distinguishing overlapping clusters. Another application of the theory to be shown is a hierarchical clustering algorithm for color image segmentation.

Acknowledgements

I would like to express my sincere thanks to Professor Shengrui Wang, my supervisor, for his many suggestions and constant support during the preparation and composition of this thesis.

I want to thanks Qingshan Jiang, Mohamed Bouguessa, and Adel Hlaoui for their advice and collaboration.

Thanks also owing to everyone on the MOdélisation en Imagerie, Vision et REseaux de neurones (MOIVRE) research team and the staff of the Département d'informatique for their assistance during my studies in Sherbrooke.

Part of the research work reported in this thesis was supported by Networks of Centers of Excellence on The Automobile of the 21st Century (NCE Auto21). During my studies at Université de Sherbrooke, I was financially supported by the Bourses Institutionnelles program of the Université de Sherbrooke and the research funds of Prof. Wang. I would like to express my sincere gratitude to all of these organizations.

Finally, I would like to express my thanks to my wife, Mei Sun for her support.

Table of Contents

List of Tables

List of Figures

Introduction

The main focus of this thesis is to investigate into two important problems in clustering: determining the optimal number of clusters and studying the phenomenon of overlapping between the clusters.

Clustering is an important research subject that has practical applications in many fields. It has been demonstrated that fuzzy clustering, using algorithms such as the Fuzzy C-Means (FCM), has clear advantages (more information for describing the relationship between objects and clusters, insensitive to initialization of cluster centers) over crisp and probabilistic clustering methods [1]. Like most clustering algorithms, however, FCM and its derivatives need the number of clusters in the given data set as one of their initializing parameters. The main goal of this thesis is to develop an effective fuzzy algorithm for automatically determining the number of clusters. We present a new algorithm for determining the number of clusters in a given data set and a new validity index for measuring the "goodness" of clustering. Experimental results and comparisons are given to illustrate the performance of the new algorithm.

The ability of a clustering algorithm to distinguish between overlapping clusters is one of the major criteria for evaluating its efficiency. However, the phenomenon of cluster overlap is still not mathematically well characterized, especially in multivariate cases. In this thesis, we study the overlapping phenomenon in the case of the Gaussian mixture, a fundamental data distribution model for many clustering algorithms. We introduce a novel concept of the ridge curve and establish a theory on the degree of overlap between two components. Based on this theory, we develop an algorithm for calculating the overlap rate. As an example, we use this algorithm to calculate the overlap rates between the classes in the IRIS data set and clear up some of the confusion as to the true number of classes in the data set. We investigate factors that affect the value of the overlap rate, and show how the theory can be used to generate truthed data as well as to measure the overlap rate of a given data set.

Finally, we give some applications of the new algorithm and the new theory. 1) We deal with a wrapper approach to the problem of feature selection for classification. Based on fuzzy clustering, we develop a new algorithm that operates by testing the error between the cluster structure of the subspace data set and the class structure of the original data set. The true number of clusters in the subspace data set introduces accurate cluster structure information. The classification error rate provides a fair evaluation on how well the subset of features represents the original feature set. 2) We propose a new approach for the objective evaluation on validity indices and clustering algorithms. We have carried out experimental studies using data sets containing clusters with various overlap rates in order to show how validity indices behave when clusters become less and less separable. 3) Based on the theory on overlapping clusters, we developed a new hierarchical algorithm for image segmentation that partially solves the problem of determining the best number of clusters. Experimental results demonstrate the effectiveness of the new algorithm.

This thesis is organized as follows:

- Introduction: summary of our research work and the organization of this thesis.

- In Chapter 1, we outline the problem of cluster analysis and the main clustering techniques, and outline the main premises of this thesis.

- In Chapter 2, the new validity index function for FCM and two new FCM-based model selection algorithms are described.

- In Chapter 3, a theory on the overlapping phenomenon is established and an algorithm for calculating the overlap rate is designed. An application of the theory to generate "truthed" data sets for evaluating validity indices is given.

- In Chapter 4, a new feature selection algorithm for classification is proposed, based on the model selection clustering algorithm in Chapter 2. On the other hand, a hierarchical algorithm, based on the overlap theory established in Chapter 3, is proposed. The performance of the hierarchical algorithm is demonstrated in an automatic color image segmentation application.

- Conclusion: we conclude this thesis and indicate future work.

Chapter 1

Major Issues in Cluster Analysis

Data analysis is the foundation of many computing applications. One key element of data analysis procedures is clustering, or the classification of measurements based on either (i) goodness-of-fit to a postulated model, or (ii) geometrical grouping revealed through analysis.

1.1 Data Clustering

Intuitively, clustering is the process of grouping the data into classes or clusters so that the objects within a cluster are highly similar and the objects in different clusters are highly dissimilar. Cluster analysis aims to organize a collection of patterns into clusters based on similarity. Clustering has its root in many fields, such as mathematics, computer science, statistics, biology, and economics. In different application domains, a variety of clustering techniques have been developed, depending on the methods used to represent data, the measures of similarity between data objects, and the techniques for grouping data objects into clusters.

The following are some examples of applications of cluster analysis:.

- In business, cluster analysis is useful in discovering distinct groups in customer bases and characterizing customer groups based on purchasing patterns [2].

- In biology, cluster analysis is used to derive plant and animal taxonomies, categorize genes with similar functionality, and gain insight into structures inherent in populations [3].

- In geography, clustering helps in the identification of areas of similar land in an earth observation database [4, 5].

- In insurance, cluster analysis is used to identify groups of automobile insurance policy holders with a high average claim cost, as well as to identify groups of houses in a city according to house type, value, and geographical location [6].

- In internet applications, cluster analysis helps grouping documents on the Web for information retrieval [7].

Cluster analysis can also be used as a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and focus on a particular set of clusters for further analysis [8]. In addition, it may serve as a pre-processing step for other algorithms, such as characterization and classification, which would then operate on the detected clusters [9].

In applications, the clustering process consists of a series of data analysis steps. A typical clustering activity involves the following steps [10]:

1. data pre-processing (including feature extraction and selection);
2. definition of a distance function for measuring the similarity between data points;
3. clustering or grouping;
4. assessment of the output

Data pre-processing involves choosing the number, type and scale of the features, which often depends on feature selection and feature extraction. Feature selection chooses important features. Feature extraction transforms input features into new salient features. They are often used in obtaining an appropriate set of features to use in clustering for avoiding the curse of dimensionality in high dimension case [11]. Another data preprocessing is removing the outliers from data. Outliers are data objects that do not comply with the general behavior or model of the data. Because outliers often lead to biased clustering results, this data must be filtered out to obtain the true clusters.

Since similarity is fundamental to the definition of a cluster, a measure of the similarity between two data objects drawn from the same feature space is essential to most clustering procedures. Because of the variety of feature types and scales, the distance measure (or measures) must be chosen carefully. It often depends on the application. It is most common to calculate the dissimilarity between two objects using a distance measure defined on the feature space. A variety of distance measures are in use in different fields [2, 10, 12]. A simple distance measure like Euclidean distance can often be used to reflect dissimilarity between two data objects. Some similarity measures, such as product-moment correlation and simple matching coefficients [13], can be used to characterize the conceptual similarity between data objects. In graph clustering, error-correcting subgraph isomorphism can be used to measure the similarity between two graphs [14].

Grouping the data objects into clusters is the main step. The data objects are grouped into clusters based on a number of different approaches. Partition-based clustering and hierarchical clustering are two of the main techniques. Partition-based clustering often starts from an initial partition and optimizes (usually locally) a clustering criterion. Hard clustering, in which each data object belongs to a single cluster, and fuzzy clustering, in which each object is assigned a degree of membership in ever cluster, are the two main partition-based techniques. Hierarchical clustering techniques generate a nested series of partitions based on a criterion, which measures the similarity between clusters or the separability of a cluster, for merging or splitting clusters. Other techniques include density-based clustering, model-based clustering and grid-based clustering. These techniques are described in detail in the next section.

Evaluating the quality of the clustering results is another important issue. Clustering is an unsupervised procedure. There is no objective criterion for evaluating the clustering results; they are assessed using a cluster validity index. In general, geometric properties, including the separation between clusters and compactness within a cluster, are often used to measure the quality. The cluster validity index also plays an important role in determining the number of clusters. It is expected that the optimal value of the cluster validity index should be obtained at the true number of clusters. A general approach for determining the number of clusters is to select the optimal value of a certain cluster validity index. Whether a cluster validity index yields the true number of clusters is a criterion for the validity index. Most existing criteria give good results for data sets with well separated clusters, but usually fail for complex data sets, for example, data sets with overlapping clusters.

1.2 Definitions and Notation

The following terms and notations are used throughout this thesis.

1. A pattern (or data object) x is a single data item used by the clustering algorithm. It typically consists of a vector of d features: $x = (x^1, ..., x^d)$.

2. d is the dimensionality of the pattern or the feature space.

3. The individual scalar components x^i of a pattern x are called features (or attributes).

4. A pattern set (or data set) is denoted by $X = \{x_1, ..., x_n\}$. The i^{th} pattern in X is denoted by $x_i = (x_i^1, ..., x_i^d)$.

5. A cluster (or class) can be viewed as a source of patterns whose distribution in feature space is governed by a probability density specific to the cluster. Clustering techniques attempt to group patterns (data) so that the clusters thereby obtained reflect the different pattern generation processes represented in the pattern set (data set).

6. The set of centers of clusters (or classes) is denoted by $V = \{v_1, ..., v_c\}$

7. A clustering technique assigns a cluster label l_k to each pattern x_k, identifying its cluster. A cluster in a data set is represented by the set of all labels for the pattern set $L = \{l_1, ..., l_n\}$, with $l_i \in \{1, ..., c\}$, where c is the number of clusters.

8. Fuzzy clustering procedures assign to each input pattern x_k a degree of membership u_{ki} in each output cluster i_{th}.

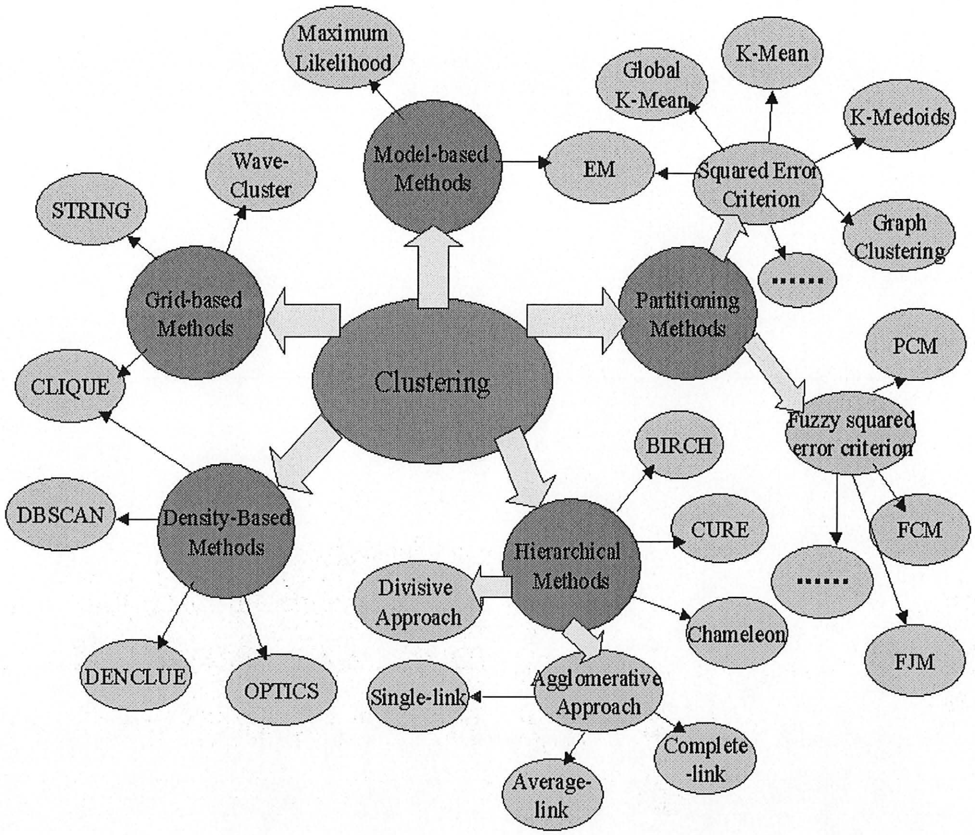


Figure 1.1: Categories of clustering techniques

1.3 Clustering Techniques

In this section, we review some of the major existing clustering techniques, focusing on the hierarchical and the partition-based approaches. Fig.1.1 describes these techniques.

1.3.1 Partition-based Clustering

For a given data set with n data objects, a partition-based clustering algorithm classifies the data into k ($k < n$) groups (clusters), which satisfy the following requirements: 1) each group (cluster) must contain at least one object; and, 2) each object must belong to exactly one group (cluster) (in fuzzy partitioning techniques, this requirement can be relaxed). The basic hypothesis of the approach is that the data fit a mixture of probability distributions of a certain type, such as Gaussian. Formally, these approaches optimize a criterion function which measures the "quality" of the clusters (including the similarity within the same cluster and dissimilarity between different clusters), defined either locally (on a subset of the partitions) or globally (over all of the partitions). In general, a partition-based clustering algorithm starts by assigning the n data objects to the k clusters. After that, it uses an iterative re-allocation technique to improve the partitioning by moving objects from one cluster to another. There are two major criterion functions that are widely used to design partition-based algorithms. One is for crisp clustering, the other is for fuzzy clustering.

1.3.1.1 The squared error criterion

The squared error criterion is one of the most intuitive and frequently used criterion function in crisp partition-based clustering techniques. For a pattern set X and a clustering L with C clusters, it is defined as follows:

$$e^2(X, L) = \sum_{i=1}^{C} \sum_{k=1}^{n_i} \|x_k^{(i)} - v_i\|^2, \tag{1.1}$$

where $x_k^{(i)}$ is the k^{th} data object belonging to the i^{th} cluster, n_i is the number of data objects in the i_{th} cluster and v_i is the centroid of the i_{th} cluster. The major algorithms based on the criterion are the K-Means (centroid-based approach) and the K-Medoids (representative object-based approach).

The K-Means is a classic partition-based clustering algorithm [15]. It is the most popular clustering algorithm. It starts from an initial partition (often random), reassigns the patterns to clusters based on the similarity between patterns and cluster centers, which are the means of the data objects in particular clusters, and recalculates the new cluster centers. The iteration continues until the squared error ceases to decrease significantly (or a certain convergence criterion is met, e.g., there is no more reassignment of any pattern from one cluster to another [12]). The K-Means algorithm is widely used because it is easy to implement, and its time complexity is $O(tkn)$, where n is the number of patterns, k is the number of clusters and t is the number of iterations. A major problem with this algorithm is that it is sensitive to the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen.

The K-Medoids [16] is another important partitioning clustering algorithm. Instead of using the mean of the objects in a cluster, the K-Medoids uses the most centrally located object in the cluster as the cluster center. It uses the same strategy as the K-means: initializing K cluster medoids, reassigning the remaining data objects to clusters based on the similarity between patterns and cluster medoids, and choosing the new medoids to decrease the squared error, until the squared error ceases to decrease significantly after some number of iterations (or no data objects are reassigned). The K-Medoids is more robust than the K-Means in the presence of noise and outliers, because a medoid is less influenced by outliers or other extreme values than a mean. However, its time cost is higher than that of the K-Means because of the time used in determining the new medoids [16]. Both kinds of methods work well with isolated, compact clusters.

There are quite a few variants of the K-Means method. These may differ in the selection of the initial k centers, the calculation of dissimilarity (particularly for data sets with categorical values, or K-Models) and the strategies for calculating cluster means (for special kind of data clustering, such as graph clustering) [14, 17, 18, 19]. Recently, some new variants have been developed focusing on global convergence and efficiency for large data sets. For example, in [20], Likas *et al.* present a global K-Means algorithm whose main idea is that an optimal solution for a clustering problem with K clusters can be obtained using a series of local searches (using the K-Means algorithm). In [21], Kanungo *et al* give an efficient K-Means algorithm based on the use of a kd-tree [22] to structure the data objects. The idea is to restructure "bad" structured clusters, which have multiple candidate centers, and maintain the "good" structured clusters, which have few candidate centers in each iteration. The efficiency of the algorithm is related to the degree of separation between clusters. The EM (Expectation Maximization) [23, 24] algorithm extends the K-Means paradigm in a different way. Instead of assigning each object to a dedicated cluster, it assigns each object to a cluster according to a weight representing the probability of membership.

1.3.1.2 The fuzzy squared error criterion

Hard clustering approaches generate partitions; in a partition, each data object belongs to one and only one cluster. So, the clusters in a hard clustering are crisp. Fuzzy clustering extends this relationship to each data object and every cluster by introducing the concept of membership [25]. The output of such an algorithm is a "fuzzy" partition. Below, we will review a fuzzy squared error criterion function and some partition-based fuzzy clustering algorithms based on the criterion.

For a given data set X with n data objects and c clusters, a fuzzy partition is defined by an $n \times C$ matrix $U = (u_{ki})$, where u_{ki} represents the degree of membership of data object x_k in cluster K_i, $u_{ki} \in [0, 1]$. The fuzzy squared error criterion function is described as follows:

$$E^2(X, U) = \sum_{k=1}^{n} \sum_{i=1}^{C} u_{ki}^m \| x_k^{(i)} - v_i \|^2, \tag{1.2}$$

where m is the fuzzifier, which is a control parameter of fuzziness.

The Fuzzy c-Means (FCM) is the most popular fuzzy clustering algorithm based on the fuzzy squared error criterion function [26, 27]. In general, it starts with an initial fuzzy partition (membership matrix U, often random). Calculating the center of each cluster and recalculating the membership matrix U are the main steps of the iteration procedure. The algorithm stops when U ceases to change significantly (controlled by a threshold). In fuzzy clustering, each cluster is a fuzzy set of all the patterns. This character lets us obtain more information about the data distribution and the cluster structure. If necessary, we can obtain a hard clustering from a fuzzy clustering by crispening the membership value. Even though it is better than the hard K-Means algorithm at avoiding local minima, FCM can still converge to local minima of the fuzzy squared error criterion [28]. FCM's major problem is its high time cost.

Some significant variants of FCM focus on modifying the criterion function. In [29], Krishnapuram and Keller presented a possibilistic approach to clustering (PCM) designed to handle the noisy environments and "thin shell" clusters, such as curves and surfaces. This approach modifies the criterion function by adding a term that forces the membership values to be as large as possible, thus avoiding trivial solutions. Moreover, PCM loosens the membership constrain of FCM, in which the sum of a given data object's in all clusters is 1, replacing it by a soft one: the sum is less than 1. Noise thus can be easily identified because of it is assigned low membership values in every cluster, totalling less than 1. In [1], Menard and Eboueya introduced an extra physical information into fuzzy criterion function. Their resulting formulae have a clearer physical meaning than those of the classical algorithms. These improvements are obtained at the cost of increased time complexity. In [30], another variant of FCM is given by Hoppner and Llawonn. The algorithm enhances the bound between clusters by restricting the memberships to 0 and 1. In [31], Romdhane *et al* gave a method for determining the fuzzifier by a heuristic scheme.

There are some other variants improving FCM by eliminating some of the clustering parameters, such as the number of clusters. In [32], C. W. Tao proposes an algorithm without a *priori* information about the number of clusters. In his algorithm, multiple centers are used to represent the non-spherical shape of clusters. Thus, it can handle non-traditional curved clusters. The high time cost is the main problem of this algorithm. In [33], A. Devilleza *et al.* present a clustering method able to divide a set of points into nonconvex classes without a priori information about their number. The two main steps of the algorithm are 1) establishing multiple sub-clusters by FCM; and 2) combining these sub-clusters into suitable clusters using a hierarchical technique. However, the method presents several limitations when clusters are overlapping. Moreover, the algorithm does not succeed in computing the number of clusters.

Another fuzzy clustering is Fuzzy J-Means (F-JM) [34], designed by Hansen *et al.* It moves the objects what belong to the neighborhood of the current solution defined by all possible centroid-to-pattern relocations. This crisp solution found is then transformed into a fuzzy one by an alternate step, i.e., by finding centroids and membership degrees for all patterns and clusters. Like the FCM method, the F-JM heuristic may be stuck in local minima. The F-JM heuristic is embedded into the variable neighborhood search (VNS) metaheuristic framework.

In most partitioning clustering algorithms, the number of clusters is an important parameter. Although there are many algorithms proposed to determine the values of this parameter. However, they are often false when the data sets with overlapping clusters and non-sphere clusters. It is still an open question in applications of this kind of clustering algorithm, especially in unsupervised application practices. Later in the chapter, we will discuss the problem determining the number of clusters, specifically for the FCM-based clustering algorithm.

1.3.2 Hierarchical Clustering

A hierarchical clustering algorithm creates a hierarchical decomposition of the given set of data objects. Depending on the decomposition approach, hierarchical algorithms are classified as agglomerative (merging) or divisive (splitting). The agglomerative approach starts with each data point in a separate cluster or with a certain large number of clusters. Each step of this approach merges the two clusters that are the most similar. Thus after each step, the total number of clusters decreases. This is repeated until the desired number of clusters is obtained or only one cluster remains. By contrast, the divisive approach starts with all data objects in the same cluster. In each step, one cluster is split into smaller clusters, until a termination condition holds. Because of the difficulty of splitting a cluster into smaller clusters (this is essentially the clustering problem itself), the agglomerative approach is more pervasive than the divisive approach. Here, we introduce some basic agglomerative and divisive algorithms.

1.3.2.1 The agglomerative approach

This is a bottom-up strategy. It often starts with a large number of clusters and stops with only one or a desire number of clusters remaining. Most hierarchical clustering methods belong to this category. The core of the agglomerative approach is determining the most similar clusters and merging them to yield a single cluster. These methods differ only in the technique used for measuring the similarity between clusters. Some widely used measures are as follows:

- Minimum distance (Single-link) [35] defines the similarity between clusters as the minimum of the distance between all pairs of data objects drawn from the two clusters (one data object from the first cluster and the other from the second). It suffers from a chaining effect, but it tends to give elongated clusters and is sensitive to noise [36].

- Maximum distance (Complete-link) [37] defines the similarity between clusters as the maximum of the distance between all pairs of data object drawn from the two clusters. Different from the single-link approach, it produces tightly bound or compact clusters, and is more computationally expensive [38]. It shows the same property as the single-link approach when dealing with noisy data sets.

- Average distance (Average-link) defines the similarity between clusters as the average distance of all pairs of data objects drawn from the two clusters. It alleviates the problems of the single- and complete- link approaches to some extent.

1.3.2.2 The divisive approach

This is a top-down strategy. It starts with all the objects in the same cluster. In each step, a cluster is split into smaller clusters according to some measure until a termination condition is satisfied. For example, in the bisecting k-means [39], the whole data set is first partitioned into two groups using the 2-Means algorithm. Then the 2-Means is applied to further split the larger of the two groups, and so on until an appropriate number of clusters are obtained. Divisive methods are less popular, even though algorithms such as the bisecting k-means are quite efficient when the number of clusters required is small, and often give on good results as well.

1.3.2.3 Some recent algorithms

Some recent algorithms are combinations of hierarchical and other techniques.

In [40], Zhang *et al.* present the BIRCH (Balanced Iterative Reducing and Clustering using Hierarches) algorithm. This algorithm is designed for very large data sets. It makes a large clustering problem tractable by concentrating on densely occupied portions, and using a compact summary. It utilizes measurements that capture the natural closeness of data. These measurements are stored and updated incrementally in a height-balanced tree. The initialization of the parameter seriously affects the efficiency of the algorithm.

In [41], Guha *et al* describe their CURE (Clustering Using REpresentatives) algorithm. It adopts a middle ground between centroid-based and representative object-based approaches. Instead of using a single center to represent a cluster, a certain number of representative points are used. The representative points of a cluster are generated by first selecting well-scattered for the cluster and then "shrinking" or move them toward the cluster center by a specified fraction "shrinking factor". At each step of the algorithm, two clusters with the closest pair of representative points are merged. Because there is more than one representative point in a cluster, CURE can find non-spherical and variable-size clusters. The shrinking of clusters reduce the effect of outliers.

In [42], Karypis *et al.* discuss the Chameleon algorithm. This algorithm uses dynamic modelling in cluster aggregation. It uses the connectivity graph corresponding to the K-nearest neighbor model sparsity of the connectivity matrix: the edges of the K most similar points to any given point are preserved, the rest are pruned. The algorithm has two stages. The first generates a large number of small tight sub-clusters using a graph partitioning. The second agglomerates these small sub-clusters based on measures of both the relative interconnectivity and the relative closeness of any pair of clusters. Thus the algorithm does not depend on a static, user-supplied model. However, the processing time cost for high-dimensional data may be $O(n^2)$ time for n objects in the worst case.

Hierarchical algorithms are more versatile than partitioning algorithms. For example, the single-link clustering algorithm works well on data sets containing non-isotropic clusters including well-separated, chain-like, and concentric clusters, whereas a typical partitioning algorithm such as the k-means algorithm works well only on data sets having isotropic clusters [36].

The time and space complexities of hierarchical algorithms are typically higher than those of partition-based algorithms [43]. As mentioned previously, measuring the similarity between clusters is the key of hierarchical algorithms. Any algorithm that uses the distance between each pair of objects to measure similarity will result in $O(n^2)$ computation and memory complexity. This is the main reason why hierarchical algorithms have higher time and space complexities. To reduce time and space complexity, the Chameleon algorithm gives a good approximation by generating sub-clusters. A useful strategy is measuring the distance between clusters based on a certain distribution model. Our work using such an approach is published [44].

1.3.3 Other Techniques

In addition to the general partition-based and hierarchical approaches, there are various approaches designed for clustering applications in specific fields. Below, we will introduce some clustering techniques for spatial data and data with arbitrarily shaped clusters.

Density-based clustering: Most partitioning methods cluster objects based on the distance between objects. These methods can find spherical-shaped clusters, and possibly elliptical (FCM), but they encounter difficulties in discovering clusters of arbitrary shape. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density (number of data objects) in a "neighborhood" exceeds some threshold: that is, for each data object within a given cluster, a neighborhood of a given radius contains at least a minimum number of objects. Such methods are useful for filtering out noise (outliers) and discovering clusters of arbitrary shape. Some of the major density-based methods are described below.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [45] is a typical density-based method that proceeds by growing cluster according to a density threshold. OPTICS (Ordering Points to Identify the Clustering Structure) [46] is a density-based method that computes an augmented clustering ordering for automatic and interactive cluster analysis. DENCLUE (Density clustering) [47] is another density clustering algorithm, which is based on a set of density distribution functions.

Grid-based clustering: Grid-based methods divide the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure (i.e. on the quantized space). The main advantage of this approach is its fast processing time, which is independent of the number of data objects and dependent on the number of cells in each dimension in the quantized space.

STRING (STatistical Information Grid) [48], a grid-based multi-resolution approach, collects statistical information in grid cells. WaveCluster(Clustering using Wavelet Transformation) [49], another multi-resolution approach, transforms the original data space in to a frequency space by a wavelet transform. CLIQUE (CLuster In QUEst) [5] is an integrated, density-based and grid-based clustering method for clustering high-dimensional data.

Model-based clustering: The underlying assumption of this approach is that the data objects in a cluster come from one of several distributions, which is often viewed as the cluster model. The goal is to estimate the parameters of each distribution and (perhaps) determine their number. Most of the work on this method has assumed that the individual components of the mixture of density are Gaussian, and in this case the parameters of the individual Gaussians are to be estimated by the procedure. Traditional approaches to this problem involve obtaining (iteratively) a maximum likelihood estimate of the parameter vectors of the component densities [10, 50, 51]. More recently, the Expectation Maximization (EM) algorithm [23, 24] (a general-purpose maximum likelihood algorithm for missing-data problems) has been applied to the problem of parameter estimation. In the EM framework, the parameters of the component densities are unknown, as are the mixing parameters, and these are estimated from the patterns. The EM procedure begins with an initial estimate of the parameter vector and iteratively rescores the objects against the mixture of density produced by these parameters. The rescored objects are then used to update the parameter estimates. In a clustering context, the scores of the objects (which essentially measure their likelihood of being drawn from particular components of the mixture) can be viewed as hints at the class of the pattern. Those objects placed (by their scores) in a particular component would therefore be viewed as belonging to the same cluster.

Some recent clustering algorithms integrate ideas from several clustering methods. It is sometimes difficult to classify a given algorithm as uniquely belonging to one clustering method. For example, Zhou provides a hybrid approach [52], which works on a hierarchical framework and takes a hybrid criterion based on both distance between clusters and density within each cluster. This hybrid method can easily identify arbitrarily shaped clusters and can be scaled up to handle very large data sets efficiently.

In summary, there are two types of clustering approaches: assignment and hierarchical. The first type assigns data objects to clusters based on a certain approach, such as partition-based clustering, density-based clustering, model-based clustering. These methods often require the number of clusters as an input parameter and have lower computational complexity. The second type creates a hierarchical cluster structure, as in hierarchical clustering and grid-based clustering. The methods in this group often have higher computational and memory complexity. The choice of clustering algorithm depends both on the type of data and on the particular purpose and application. If cluster analysis is being used as a descriptive or exploratory tool, it is possible to try several algorithms on the same data to see what the data may disclose.

1.4 Important Issues Related to Cluster Analysis

In this section, we discuss some important topics about cluster analysis and briefly introduce how this thesis addresses the underlying issues.

1.4.1 Evaluating the Clustering Results

Evaluating the results of clustering (or pre-clustering) is another important topic. Because clustering is an unsupervised procedure, clustering results need be judged by an external criterion. For low dimensional data sets (1-, 2- or 3-dimensional), humans can also evaluate the clustering results by visual observation. In general, the evaluation is often based on a clustering validity index. Depending on the type of clustering approach (crisp or fuzzy), there are various validity indices designed for evaluating the clustering results [53]. Most of them are based on intuitive knowledge about the geometric properties of clusters, which includes the separation between clusters and compactness within a cluster. Many of these validity indices work well in cases where the clusters are well separated. However, because they disregard lack of considering the theoretical characterization of the overlapping phenomenon, they often yield questionable results for cases involving overlapping clusters [54]. A theoretical understanding of the phenomenon of overlapping will help us in distinguishing overlapping clusters, and hence, in correctly determining the number of clusters and understanding the cluster structure of a given data set.

1.4.2 Determining the Number of Clusters

The number of clusters is the most important parameter in many algorithms (e.g. partition-based and model-based clustering methods). These algorithms require the user to input parameters (for example, the number of clusters and the initializing cluster centers). The clustering results can be quite sensitive to the input parameters. A lot of work has been done on determining the optimal number of clusters [10, 4, 50, 55] and success in many applications. However, there are no completely satisfactory methods for determining the number of population clusters for any type of cluster analysis [8, 53, 56].

An intuitive method for determining the number of clusters is to pre-cluster the data set into k clusters, where k is a possible value of the number, and evaluate the result using a cluster validity index. The pre-clustering often uses a basic clustering algorithm, such as the K-Means or FCM. The cluster validity index provides a numeric criterion for the possible number of clusters, k, by evaluating the clustering result. The optimal value of the index should be obtained at the true number of clusters.

1.4.3 Cluster Overlapping

The next important topic is cluster overlapping. The phenomenon of cluster overlapping is present in real applications. Many algorithms fail to distinguish overlapping clusters. One relative study is to measure the distance between clusters. Based on certain models, the distance describes the characteristic of separation between distributions (components). However, when the components in a mixture are overlapped, the inbeing of these components will deviate. For example, the configuration of the probabilistic density function of the mixture will exchange. The level of the variation is based on the degree of overlapping between the components. How to describe the properties of component overlapping is the goal of the topic. Recently, quite a few researchers have obtained significant results on the overlapping phenomenon in lower-dimensional cases [57, 58], based on Gaussian mixture models. More investigation is needed for the multi dimensional case.

1.4.4 Feature Selection

Feature selection is a very important step in a classification system or data mining system. Many clustering algorithms work well on data sets that contain low-dimensional data objects. However, a large data set may contain millions of high-dimensional objects. Clustering a large data set may lead to biased results. One simple reason is that high-dimensional data tend to be more separated than low-dimensional data. The cluster structure of these types of data sets is not clear. In practice, features are often interdependent and relative. Finding the subset of features in which features are non-correlated each other and have a clear cluster structure is the important for understanding inherent relation between variables in the data set and/or for characterizing different classes of objects involved.

1.5 Contribution of This Thesis

In view of the above discussion, in this thesis, we focus on two issues in cluster analysis: determining the optimal number of clusters and distinguishing overlapping clusters.

1.5.1 Determining the Number of Clusters in a Given Data Set

A common approach for determining the number of clusters is an iterative trial-and-error process, which performs the model selection according to the terms used by Jain [10]. The approach involves running the clustering algorithm with different initial values for the number of clusters and comparing results in order to determine the most appropriate number of clusters. The two main steps in this "trial-and-error" method are pre-clustering, which pre-groups the data objects into k clusters (here, k is a possible value for the real number of clusters), and judging, which exams the pre-clustering results. For the first step, we use the Fuzzy C-Means (FCM) as the basic algorithm for pre-clustering. As mentioned above, FCM is able to perform membership grading, which gives more helpful information for taking advantage of the relationship between data objects and clusters. This property is important for comparing the clustering results obtained with different initial numbers of clusters.

The step of judging the pre-clustering results is carried out based on a validity function (index) that measures the internal cohesion within each cluster and the external separation between clusters. It judges not only the clustering results, but also the quality of some fundamental parameters of the clustering algorithm (for example, the number of clusters and the initialization for prototypes in partitioning methods). A good validity index should perform in such a way that its optimal value is reached only when the correct (true) values of the fundamental parameters are used and true clusters are obtained.

This thesis makes two contributions to the determination of the number of clusters. First, we propose a new validity index, based on measuring both the compactness within each cluster (or actually, its inverse, scatter) and the separation between clusters (distance). The experiments demonstrate the advantage of the new index in differentiating overlapped clusters. Second, based on the new validity index and the FCM algorithm, we propose a new algorithm for automatically determining the number of clusters. The new algorithm improves the conventional model selection process by reducing the randomness in the initialization of each pre-clustering phase. The experiments show the superiority of the new algorithm in both computational complexity and stability.

1.5.2 Theory Regarding Component Overlap in Mixture Model

Distinguishing overlapped clusters is an important and difficult task in clustering. Some authors have focused on this topic and have obtained significant results in lower dimensional cases. Qu *et al.* [57] have conducted simulations in order to reveal the relationship between overlapping components and the distance between two adjacent components in a Gaussian mixture. Aitnouri *et al.* [58] have given a formal definition of the overlap rate and proposed two algorithms for generating data sets with controlled overlap rate. Both studies apply to the 1-D case. Indeed, due to its complexity the overlapping phenomenon in the multidimensional case requires minute investigation.

This thesis reports our study on the overlapping phenomenon in the multivariate case, based on a Gaussian mixture model. First, we establish a theory on measuring the rate of overlap between two components in a mixture model based on a series of theorems. Second, we investigate the overlap rate of some published data sets with overlapping classes. Third, we investigate the factors that affect the value of the overlap rate.

1.5.3 Applications of Our New Algorithms

This thesis also report some applications of our new algorithms for determining the number of clusters in feature selection and hierarchical clustering in color image segmentation. First, we propose a wrapper approach to feature selection using the new clustering technique mentioned above. The approach is based on the fact that the selected feature subset is "structurally similar" to the original feature set. Based on the efficient clustering algorithm, we design a novel algorithm for feature selection by focusing on the structural similarity in the selection process. We define a classification error rate for evaluating the subset of features. Extensive test results derived by applying the new algorithm to two artificial data sets and a series of real-world data sets are reported. Second, we develop a new hierarchical algorithm for image segmentation that focuses on measuring the similarity between pairs of objects and merging the closest two. The mixture of Gaussian is taken as the basic hypothesis about the distribution of the RGB values in a color image. The overlap rate is used to measure the similarity between two objects (Gaussian components). The experimental results show the effectiveness of the new method.

Chapter 2

FCM-Based Model Selection Algorithms for Determining the Number of Clusters

Because of the important place of determining the number of clusters in cluster analysis, there have been many methods had been designed for estimating the number of clusters: a good summary is given by Gordon [59]. The "trial-and-error" [10] process is the most widely used approach for the model selection problem. The approach is often based on a basic clustering algorithm and a validity index for evaluating the clustering results. The basic clustering algorithm is used to pre-clustering data set for a supposed number of clusters. The validity index is used for evaluating the pre-clustering results. The idea is to maximize or minimize a validity index over the number of clusters. Here, we use the Fuzzy C-Means (FCM) as the basic algorithm, because the fuzzy membership can provide more information to be used. In this chapter, we report our work on determining the number of clusters: a new validity index and a new algorithm for the model selection based on the "trial-and-error" approach.

2.1 Model Selection Algorithm

In this section, we briefly introduce the basic FCM algorithm and the general model selection algorithm for determining the number of clusters in a data set.

2.1.1 FCM Algorithm

The FCM algorithm dates back to 1973 [26]. FCM-based algorithms are the most widely used fuzzy clustering algorithms in practice. The basic FCM algorithm can be formulated as follows

$$Minimize\ J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}^m \|x_k - v_i\|^2, \tag{2.1}$$

where n is the total number of data vectors in a given data set and c is the number of clusters; $X = \{x_1, x_2, \cdots, x_n\} \subset R^d$ and $V = \{v_1, v_2, \cdots, v_c\} \subset R^d$ are the feature data and cluster centers; and $U = (u_{ki})_{n \times c}$ is a fuzzy partition matrix composed of the membership of each feature vector x_k in each cluster i, where u_{ki} should satisfy $\sum_{i=1}^{c} u_{ki} = 1$ for $k = 1, 2, \ldots, n$, $u_{ki} \geq 0$ for all $i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, n$. The exponent $m > 1$ in $J_m(U,V)$ (Equation 2.1) is a parameter, usually called a fuzzifier. To minimize $J_m(U,V)$, the cluster centers (prototypes) v_i and the membership matrix U need to be computed according to the following iterative formula:

$$u_{ki} = \begin{Bmatrix} \begin{cases} \left(\sum\limits_{j=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{2}{m-1}} \right)^{-1} & if\ \|x_k - v_j\| > 0, \forall j. \\ 1 & if\ \|x_k - v_i\| = 0 \\ 0 & if\ \exists\, j \neq i\ \|x_k - v_j\| = 0 \end{cases} & \begin{matrix} k = 1, \ldots, n \\ i = 1, \ldots, c \end{matrix} \end{Bmatrix} \tag{2.2}$$

$$v_i = \frac{\sum\limits_{k=1}^{n} u_{ki}^m\, x_k}{\sum\limits_{k=1}^{n} u_{ki}^m} \quad i = 1, 2, \cdots, c \tag{2.3}$$

The basic FCM algorithm is as follows.

Algo1: Basic FCM algorithm

1. Input the number of clusters c, the fuzzifier m and the distance function $\|\cdot\|$.

2. Randomly initialize the cluster centers $v_i^0 (i = 1, 2, \ldots, c)$.

3. Calculate $u_{ki} (k = 1, 2, \ldots, n; i = 1, 2, \ldots, c)$ using Eq.(2.2) .

4. Calculate $v_i^1 (i = 1, 2, \ldots, c)$ using Eq.(2.3) .

5. If $\max\limits_{1 \leq i \leq c} (\|v_i^0 - v_i^1\| / \|v_i^1\|) \leq \varepsilon$ then go to *Step 6*; else let $v_i^0 = v_i^1 (i = 1, 2, \ldots, c)$ and go to *Step 3.*

6. Output the clustering results: cluster centers $v_i^1 (i = 1, 2, \ldots, c)$, membership matrix U and, in some applications, the elements of each cluster i, i.e., all the x_k such that $u_{ki} > u_{kj}$ for all $j \neq i$.

7. Stop.

2.1.2 Determination of the Number of Clusters

The following model selection algorithm applies the basic FCM clustering algorithm to the data set for $c = C_{\min}, ..., C_{\max}$ and chooses the best value of c based on a (cluster) validity criterion. Here, $C_{\min}$ and $C_{\max}$ are predefined values that represent, respectively, the minimal and maximal numbers of clusters between which an optimal number is sought.

Algo2: FCM-based model selection algorithm

```
1. Choose $C_{\min}$ and $C_{\max}$.
2. For $c = C_{\min}$ to $C_{\max}$:
    2.1. Initialize cluster centers ($V$).
    2.2. Apply the basic FCM algorithm to update the membership matrix ($U$) and the cluster centers ($V$).
    2.3. Test for convergence; if no, go to 2.2.
    2.4. Compute a validity value $V_d(c)$.
3. Compute $c_{Opt}$ such that the cluster validity function $V_d(c_{Opt})$ is optimal.
```

Several techniques exist for initializing cluster centers (Step 2.1). Random initialization is often used because of its simplicity. Other initialization methods could be used in many cases. Recently, an empirical comparison of four initialization methods for the K-Means algorithm was reported in [60]. According to this study, random initialization is one of the best methods as it makes the K-Means algorithm more effective and less dependent on initial clustering and order of instances. Although it is not clear if these results can be generalized to the case of FCM, it is still reasonable to assume that random initialization is a good choice for **Algo2**.

2.2 Validity Indices for Fuzzy Clustering

The index $V_d(c_{Opt})$ in **Algo2** measures the *goodness* of the results of a clustering algorithm. A partition is considered good if it optimizes two conflicting criteria. One of these is related to within-class scattering, which needs to be minimized; the other to between-class scattering, which needs to be maximized. The major validity indices for fuzzy clustering found in the literature are reviewed in the following section. Then, the new index is introduced and the performance of the various indices is compared.

2.2.1 Existing Validity Indices

There are a number of cluster validity indices available. Some of them use only the membership values of a fuzzy partition of the data (membership matrix), others use the original data and computed cluster centers as well as the membership matrix. Here are some of the indices most frequently referred to in the literature.

- Partition coefficient V_{PC}:

$$V_{PC}(U,c) = \frac{1}{n}\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}^2 \tag{2.4}$$

- Partition entropy V_{PE}:

$$V_{PE}(U,c) = -\frac{1}{n}\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ki}\log(u_{ki}) \tag{2.5}$$

V_{PC} and V_{PE} [53] are two simple indices that are computed using only the elements of the membership matrix. For V_{PC}, a larger value means better clustering, since in this case u_{ki} tends to be closer to one or zero, so the data set is well divided. For V_{PE}, a smaller value means better clustering because of lower entropy. Both indices are easy to compute. They are useful when the data contains only a small number of well-separated clusters. However, there is a lack of direct connection to the geometrical properties of the data. Both of them seem to not handle the data well when there is overlapping clusters in it.

The indices below are some more used indices that make explicit use of data and cluster centers. Fakuyama and Sugeno proposed another validity index, which measures the discrepancy between the compactness and separation of clusters [61]. Xie and Beni defined a well-known validity index, which measures the overall average compactness against separation of the c-partition [62].

- Fakuyama-Sugeno validity V_{FS}:

$$V_{FS}(U,V,c) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ki}^{m}(\|x_k - v_i\|^2 - \|v_i - \overline{v}\|^2) \tag{2.6}$$

- Xie-Beni validity V_{Xie}:

$$V_{Xie}(U,V,c) = \frac{\sum\limits_{k=1}^{n}\sum\limits_{i=1}^{c} u_{ki}^{m}\|v_i - x_k\|^2}{n \times \min\limits_{i \neq j} \|v_i - v_j\|} \tag{2.7}$$

where $\overline{v} = \frac{1}{n}\sum\limits_{i=1}^{n} x_i$.

$V_{FS}(U,V,c)$ measures the discrepancy between compactness and separation. The first term in brackets, $\sum\limits_{i=1}^{c}\sum\limits_{k=1}^{n} u_{ki}^{m}\|x_k - v_k\|^2$, measures the compactness of the clusters while the second one, $\sum\limits_{i=1}^{c}\sum\limits_{k=1}^{n} u_{ki}^{m}\|v_i - \overline{v}\|^2$, measures the distances of the clusters representatives. It is clear that for compact and well-separated clusters we expect small values for V_{FS}. On the other hand, $V_{Xie}(U, V, c)$ is the ratio of within-cluster compactness and between-cluster separation. The function $J = \frac{1}{n}\sum\limits_{k=1}^{n}\sum\limits_{i=1}^{c} u_{ki}^{m}\|v_i - x_k\|^2$ measures the within-cluster compactness; its value will be small if the clusters are compact. $J_{min} = \min\limits_{i \neq j}\|v_i - v_j\|$ measures the separation between clusters; its value tends to be large if the clusters are well separated. $V_{Xie}(U, V, c)$ is a trade-off between compactness and separation. To obtain good clustering results, $V_{Xie}(U, V, c)$ needs to be minimized.

Recently, several other validity indices have been proposed, using different definitions of compactness and separation. Rhee and Ho (1996) proposed two indices in their paper [63], V_{RH} (Eq. 2.8. Both of their indices give better results in terms of the accuracy of the final number of clusters. However, the computational complexity for calculating each value of each of these indices is $O(n^2c)$, where n is the number of data vectors and c is the number of clusters. Zahid, *et al.* [64] proposed an index V_{ZLE}, which considers the geometrical properties, the degree of fuzzy membership and the structure of the data. Rezaee, *et al.* [65] proposed a validity index, V_{RLR} (Eq. 2.9) which depends on a linear combination of the average scattering (compactness) of clusters and distance (separation) between clusters: $\alpha \times Scattering + \beta \times Distance$. Our new index was based on the index.

- Rhee-Ho validity V_{RH}:

$$V_{RH}(U, V, c) = \frac{\frac{2}{n(n-1)}\sum\limits_{j=1}^{n-1}\sum\limits_{k=j+1}^{n}\sum\limits_{i=1}^{c} d^2(x_j, x_k)\omega_1(i, j, k)}{\frac{1}{n^2}\sum\limits_{j=1}^{n-1}\sum\limits_{k=j+1}^{n} d^2(x_j, x_k)\omega_2(j, k)} \tag{2.8}$$

where $\omega_1(i, j, k) = \min\{u_{ij}, u_{ik}\}$ and $E = \frac{2}{n(n-1)}\sum\limits_{j=1}^{n-1}\sum\limits_{k=j+1}^{n}\sum\limits_{i=1}^{c} d^2(x_j, x_k)\omega_1(i, j, k)$ is an average intra-class distance used as a measure of the compactness of the fuzzy c-partition.

$\omega_2(j,k) = \min\{\max_{i_1} u_{i_1 j}, \max_{i_2 \neq i_1} u_{i_2 k}\}$ and $D = \frac{1}{n^2}\sum_{j=1}^{n-1}\sum_{k=j+1}^{n} d^2(x_j, x_k)\omega_2(j,k)$ is an average inter-class distance for measuring the separation of the fuzzy c-partition. The maximum of $V_{RH}(U,V,c)$ corresponds to the best clustering.

- Rezaee-Letlieveldt-Reiber validity V_{RLR}:

$$V_{RLR}(U,V,c) = \alpha \times Scat(c) + Dis(c) \tag{2.9}$$

where $\alpha = Dis(C_{max})$ and c, the number of clusters, must be between C_{min} and C_{max} ($2 \leq C_{min} < C_{max} < n$), C_{min} and C_{max} being the minimum number of clusters and the maximum number of clusters, respectively. The average scattering is defined as $Scat(c) = \frac{\frac{1}{c}\sum_{i=1}^{c}\|\sigma(v_i)\|}{\|\sigma(X)\|}$, where $\sigma(X) = \{\sigma(X)^1, \sigma(X)^2, \cdots, \sigma(X)^d\}^T$, $\overline{x} = \frac{1}{n}\sum_{k=1}^{n} x_k$, $\sigma^2(X)^p = \frac{1}{n}\sum_{k=1}^{n}(x_k^p - \overline{x}^p)^2$, $\sigma(v_i) = \{\sigma(v_i)^1, \sigma(v_i)^2, \cdots, \sigma(v_i)^s\}^T$, and $\sigma(v_i)^p = \frac{1}{n}\sum_{k=1}^{n} u_{ki}(x_k^p - v_i^p)^2$, for $p = 1, 2, \cdots, d$. The distance function is defined as $Dis(c) = \frac{D_{max}}{D_{min}}\sum_{i=1}^{c}\left(\sum_{j=1}^{c}\|v_i - v_j\|\right)^{-1}$, where $D_{min} = \min_{i \neq j}\|v_i - v_j\|(i, j \in [1, c])$, $D_{max} = \max_{i,j}\|v_i - v_j\|(i, j \in [1, c])$. $Scat(c)$ indicates the compactness of the partition. A small value of $Scat(c)$ means that, on average, the clusters are compact relative to the variance of the data set. $Dis(c)$ indicates the total scattering (separation) between the clusters. The weighting factor $\alpha = Dis(C_{max})$ is introduced to compensate for differences in the scales of $Dis(c)$ and $Scat(c)$. The minimum value of V_{RLR} is believed to correspond to the best clustering.

2.2.2 A New Validity Index

In our experiments, we found that the currently used validity indices behave poorly when clusters overlap each other. This motivated our search for an efficient validity function. The validity index we propose $V_{WSJ}(U,V,c)$ has the following form:

$$V_{WSJ}(U,V,c) = Scat(c) + \frac{Sep(c)}{Sep(C_{\max})} \tag{2.10}$$

Here $Scat(c)$ is defined in the same way as in the Rezaee-Letlieveldt-Reiber index. It represents the compactness of the obtained clusters. The value of $Scat(c)$ generally decreases when c increases because the clusters become more compact. The range of $Scat(c)$ is between 0 and 1. The term representing the separation between clusters is defined as $Sep(c) = \frac{D^2_{max}}{D^2_{min}} \sum_{i=1}^{c} \left(\sum_{j=1}^{c} \|v_i - v_j\|^2 \right)^{-1}$, where $D_{min} = \min_{i \neq j} \|v_i - v_j\|$ and $D_{max} = \max_{i,j} \|v_i - v_j\|$.

To gain some insight into this definition of separation, $Sep(c)$ can be written approximately as $Sep(c) \doteq \frac{c}{c-1} \frac{D^2_{max}}{D^2_{min}} E[\frac{1}{d_c^2}]$ ($E[X]$ means the expectation of X), where d_c is the average distance from a cluster center to all the other cluster centers. Both $\frac{D^2_{max}}{D^2_{min}}$ and $E[\frac{1}{d_c^2}]$ in $Sep(c)$ are influenced by the geometry of the cluster centers. Both factors tend to be small when the cluster centers are well separated. For example, when the cluster centers form a tetrahedron, $\frac{D^2_{max}}{D^2_{min}}$ reaches its minimum, which is 1; while $E[\frac{1}{d_c^2}]$ also reaches its minimum, $\frac{1}{D^2_{max}}$, whose value depends only on the absolute distance (which is a scale factor) between any two centers. When the distribution of cluster centers is irregular, both $\frac{D^2_{max}}{D^2_{min}}$ and $E[\frac{1}{d_c^2}]$ become larger. However, their values tend to evolve in different ways when the number of clusters c in the clustering algorithm increases. In fact, $\frac{D^2_{max}}{D^2_{min}}$ will likely increase as more (calculated) cluster centers tend to result in increased D_{max} and decreased D_{min} at the same time. On the other hand, $E[\frac{1}{d_c^2}]$ will likely become more stable as the estimate of the average distance d_c becomes more accurate. This is also why $\frac{D^2_{max}}{D^2_{min}}$ is important, since it is the main factor that penalizes model structures with too many clusters.

A trade-off needs to be made between cluster scattering and cluster separation in the index. Since the value of $Sep(c)$ depends on the absolute distances between cluster centers, or in other words, it depends on a scale factor from the input data, the term $\frac{Sep(c)}{Sep(C_{\max})}$ is utilized to scale down the value of $Sep(c)$ into the same range as $Scat(c)$. A coefficient could be used to modulate the contribution of each of the two terms in $V_{WSJ}(U, V, c)$. Our experience indicates that this is not necessary. In fact, for the various data sets tested, the expression for $V_{WSJ}(U, V, c)$ given in Equation (2.10) has proved to yield more accurate results than any other index tested (see the next section). A cluster number which minimizes $V_{WSJ}(U, V, c)$ is considered to be the optimal value for the number of clusters present in the data.

2.3 A New FCM-based Clustering Algorithm

In **Algo2**, we use random initialization at the beginning of each clustering phase. By doing so, we try to ensure that the selection process is carried out under relatively general conditions and the results are as replicable as possible. However, it is easy to imagine that re-initialization at each phase could lead to computational inefficiency. Use of the clustering results obtained in previous phases may lead to a better initialization. In this section, we propose strategies that yield a new FCM-based clustering algorithm. First we explain the major steps of the algorithm in detail. Experimental results and discussions follow in subsequent sections

The **FCM-Based Splitting Algorithm (FBSA)** described below is called a splitting algorithm because it operates by splitting the "worst" cluster at each stage in testing the number of clusters c from C_{min} to C_{max}. The major differences between this algorithm and **Algo2** in the previous section lie in the initialization of cluster centers, the validity function used and the process for splitting "bad" clusters. The general strategy adopted for the new algorithm is as follows: at each step of the new algorithm (**FBSA**), we identify the "worst" cluster and split it into two clusters while keeping the other $c-1$ clusters.

Algorithm FBSA:

```
1. Choose $C_{min}$ and $C_{max}$.
2. Initialize $C_{min}$ cluster centers ($V$).
3. For $c = C_{min}$ to $C_{max}$:
    3.1. Apply the basic FCM algorithm to update the membership matrix ($U$) and
         the cluster centers ($V$).
    3.2. Test for convergence of $V$; if no, go to 3.1.
    3.3. Compute a validity value $V_d(c)$.
    3.4. Compute a score $S(i)$ for each cluster; split the worst cluster.
4. Compute $c_{Opt}$ such that the result of cluster validity function $V_d(c_{Opt})$ is optimal.
```

The general idea in the splitting algorithm **FBSA** is to identify the "worst" cluster and split it, thus increasing the value of c by one. Our major contribution lies in the definition of the criterion for identifying the "worst" cluster. In this thesis, we propose a "score" function $S(i)$ associated with each cluster i, as follows:

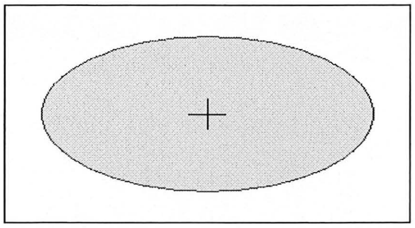


Figure 2.1: Cluster before the split. The center of the cluster is marked by "+".

$$S(i) = \frac{\sum_{k=1}^{n} u_{ki}}{n_i}$$

where n_i is the number data objects in i^{th} cluster.

In general, when $S(i)$ is small, cluster i tends to contain a large number of data vectors with low membership values. The lower the membership value, the farther the object is from its cluster center. Therefore, a small $S(i)$ means that cluster i is large in volume and sparse in distribution. This is the reason why we choose the cluster corresponding to the minimum of $S(i)$ as the candidate to split when the value of c is increased. On the other hand, a larger $S(i)$ tends to mean that cluster i has a smaller number of elements and exerts a strong "attraction" on them.

In order to split the cluster at Step 3.4 of **FBSA**, we have adapted the "Greedy" technique [66]. The "Greedy" technique aims to initialize the cluster centers as far apart from each other as possible. In an iterative manner, the "Greedy" technique selects as a new cluster center the data vector which has the largest total distance from the existing cluster centers. Adaptation of the technique for cluster splitting yields the following algorithm:

1. Identify the cluster to be split (first part of 3.4). Supposing that the cluster number is i_0, its center and the set of all the data in the cluster are denoted by V_{i_0} and E.

2. Search in E for the data vector not labelled "tested" which has the maximal total distance from all of the remaining $c-1$ cluster centers. This data vector is denoted by V_{i_1}.

3. Partition E into E_0 and E_1 based on the distance of each data vector from V_{i_0} and V_{i_1}. If $|E_1|/|E| > 10\%$, then V_{i_1} is taken as the c^{th} cluster center; else label V_{i_1} "tested" and go to Step 2.

4. Search E for the data vector not labelled "tested" which has the maximal total distance from all of the c cluster centers. This data vector is denoted by V_{i_2}.

5. Partition E into E_1 and E_2 based on the distance of the data vector from V_{i_1} and V_{i_2}. If $|E_2|/|E| > 10\%$, then V_{i_2} is taken as the $(c+1)^{th}$ cluster center, else label V_{i_2} "tested" and go to Step 4.

This algorithm ensures that the two new centers V_{i_1} and V_{i_2} are as far apart as possible from each other and from the $c-1$ centers (of the unsplit clusters). In addition, a significant number of data vectors (10% of E) are required to be in the neighborhood of each center so as to minimize the possibility of picking up an outlier. Figures 2.1 and 2.2 illustrate a typical result of the splitting algorithm.

2.4 Experimental Results

In this section, we present the performance of the proposed algorithm using validity index in Eq. 2.10. We report the experimental results for four data sets, the first one from

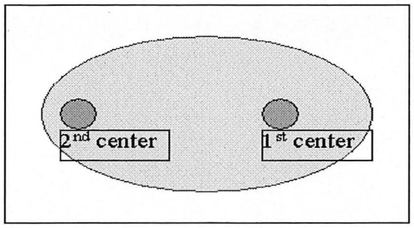


Figure 2.2: Cluster after the split. The small circles here mark the initial centers of new clusters.

the public domain, the next two generated using mixtures of Gaussian distributions and the fourth one from a real survey data set. For each of the first three data sets, we evaluate the algorithm and index using three criterions: accuracy of clustering results, stability across different runs and time cost. The fourth data set is used only for evaluating the time efficiency of the proposed algorithm.

In all experiments, the fuzzifier m in the algorithm was set to 2, the test for convergence in the basic FCM algorithm (**Algo1**) was performed using $\varepsilon = 0.001$, and the distance function $\|\cdot\|$ was defined as Euclidean distance. Choosing the best range of the number of clusters is a difficult problem. Here we adopted Bezdek's suggestion: $C_{\min} = 2$ and $C_{\max} = \sqrt{n}$ [53]. For determination of the number of clusters, the validity indices V_{PC}, V_{PE}, V_{Xie}, V_{FS}, V_{RH}, V_{ZLE} and V_{RLR} were compared with V_{WSJ}. The initialization of cluster centers in **FBSA** (step 2) and **Algo2** (step2.1) used the random procedure.

	Cluster1	Cluster2	Cluster3	Cluster4	Cluster5
x	1.0	1.0	-0.5	-0.5	-0.5
y	0.3	-0.3	0	-0.3	0.3
z	0.0	0.0	0.5	0.0	0.0

Table 2.1: Means of DataSet2

2.4.1 Data Sets

The first data set, DataSet1, is IRIS data set [67], widely used for testing clustering algorithms. This is a biometric data set consisting of 150 measurements belonging to three flower varieties. The data are represented as vectors in a 4-dimensional measurement space, in which four variables are length and width of both petal and sepal. The set consists of three classes, each of which contains 50 observations. In fact, of the three classes, two are overlapping. Halgamuge and Glesner [68] have shown that a very good classification can be obtained using only two features (petal length and petal width). In [65], Rezaee *et al.* indicate that for their index V_{RLR}, only one feature (petal length) is used to obtain the best number of classes, which is 3.

DataSet2 was generated using a mixture of Gaussian distributions. This data set is 3-dimensional and contains 5 Gaussian components (clusters). There are 50 data vectors in each of the five clusters. For each component, the three variables are independent of each other and their variance is 0.2. The means of the five clusters (components) are given in Table 2.1. Figure 2.3 shows the 3D picture. In the data set, Cluster1 and Cluster2 strongly overlap each other and Cluster3, Cluster4 and Cluster5 strongly overlap each other.

DataSet3 was also generated using a mixture of Gaussian distributions too. This example contains 500 2-dimensional data vectors. It consists of 10 Gaussian components

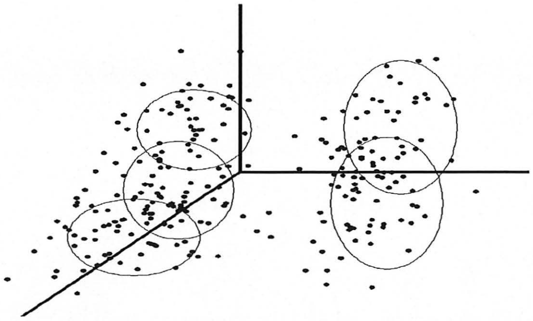


Figure 2.3: DataSet2 is a 3D data set and has 5 clusters

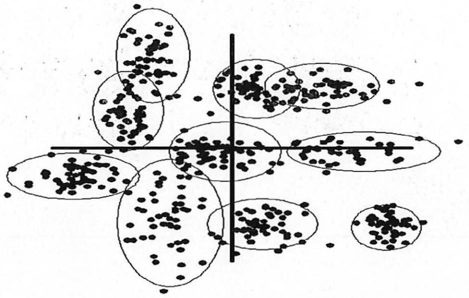


Figure 2.4: DataSet3 is a 2D data set and has 10 clusters

(clusters). For each component, the two variables are independent. There are 50 data vectors in each of the ten clusters. The means and variances of the 10 clusters (components) are given in Table 2.2. This data set has been generated so that Cluster1 and Cluster2 overlap, and Cluster3 and Cluster4 also overlap. Figure 2.4 illustrates the data set.

		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
x	mean	0.0	4.5	4.5	1.5	-2.0	-5.5	-3.5	-2.5	2.0	7.0
	variance	1.0	1.5	2.0	1.0	1.0	1.0	0.5	0.5	0.5	0.5
y	mean	0.0	3.0	0.0	3.0	-3.0	-1.0	2.0	4.5	-3.5	-3.5
	variance	0.5	0.5	0.5	0.5	1.5	0.5	1.0	1.0	1.0	0.5

Table 2.2: Means and variances of DataSet3

2.4.2 Comparison of Accuracy of Clustering Results

The main objective of this subsection is to compare the performance of different validity indices in determining the true number of clusters. There are two parts in this comparison: the optimal number of clusters(c_{Opt}) and the errors between cluster centers and component means. Normally, each run of the algorithm **Algo2** or the algorithm **FBSA** involves only one validity index. However, random initialization in these algorithms may have some effect on their (average) performance given that the number of runs of each algorithm is always limited (to 20 in our experiments). In order to eliminate the disparities in performance induced by random initialization, we simply need to compute all the validity indices on the same set of the clusters, yielded by **Algo2** or **FBSA** for $c = C_{min}$ to C_{max}, and record and compare the optimal cluster number corresponding to each validity index. For this reason, all the experiments in this section were performed with all the tested validity indices implemented within **Algo2** and **FBSA**. With these settings, if two validity indices yield the same optimal number of clusters in a run, they yield exactly the same clusters too. This point is particularly important for understanding the results discussed in later of this subsection, where two validity indices can yield exactly the same average position error for cluster centers even with a significant number of random initializations in the clustering algorithms.

2.4.2.1 Accuracy of the optimal number of clusters(c_{Opt})

Finding the true number of clusters is a fundamental goal of the clustering algorithm. The validity function often plays a key role in model selection approaches. Here we have tested the existing well-known validity indices V_{PC}, V_{PE}, V_{Xie}, V_{FS}, V_{RH}, V_{ZLE} , V_{RLR} and our new validity index V_{WSJ} in **Algo2** and **FBSA.** Each algorithm was run 20 times with different initial centers in order to evaluate the stability of the algorithm and the validity index used.

Tables 2.3, 2.4 and 2.5 give the results for the optimal number of clusters (20 runs) when all validity indices are applied to **Algo2** for the three data sets. In these tables, $c_{Opt}(m)$ means that the optimal value c_{Opt} was obtained m times in 20 runs. For the IRIS data set, V_{WSJ} yields an optimal number of clusters of 3 (the best number of clusters) in 19 out of 20 runs. However, few of the existing validity indices result in the true cluster number.

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
run1~ run20	2(20)	2(20)	2(20)	3(1),4(3), 5(8),6(6),8(2)	4(1),5(4), 6(10),7(4),8(1)	2(20)	2(14),3(6)	3(19),5(1)
accuracy rate	0/20	0/20	0/20	1/20	0/20	0/20	6/20	19/20

Table 2.3: The optimal number of clusters of Algo2 for DataSet1

For DataSet2, the very strong overlapping of clusters and the random initialization procedure result in a very serious consequence: no validity indices often yield the true cluster number. Nevertheless, V_{WSJ} produces the correct number of clusters in 8 runs out of 20.

Most of the validity indices divide the data set into two clusters, one of which results from the merging of two Gaussian components and the other from the merging of three Gaussian components. We notice that V_{WSJ} and V_{FS} sometime produce 4 clusters. In this case, two clusters are merged and the other three are kept. Compared with others, the results produced by these two validity indices are acceptable.

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
run1~ run20	2(20)	2(20)	2(20)	3(9),4(8), 5(2),6(1)	2(17),3(3)	2(20)	2(20)	2(8), 4(4),5(8)
accuracy rate	0/20	0/20	0/20	2/20	0/20	0/20	0/20	8/20

Table 2.4: The optimal number of clusters of Algo2 for DataSet2.

For DataSet3, it contains 10 clusters and the degree of spread within clusters is different. There are four components tending to merge into two. This type of data set is difficult to identify. V_{Xie}, V_{FS} and V_{WSJ} show the best identification ability for the data, reaching a 90% accuracy rate.

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
run1~ run20	2(20)	2(20)	7(1),9(1), 10(18)	10(18), 11(1),14(1)	10(8),11(1) 13(2),19(2) 21(5),23(2)	2(20)	5(19) 7(1)	7(1),9(1), 10(18)
accuracy rate	0/20	0/20	18/20	18/20	8/20	0/20	0/20	18/20

Table 2.5: The optimal number of clusters of Algo2 for DataSet3

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
run1~run20	2	2	2	6	7	2	2	3
correct rate	0/20	0/20	0/20	0/20	0/20	0/20	0/20	20/20

Table 2.6: The optimal number of clusters of FBSA for DataSet1

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
run1~run20	2	2	2	4	2	2	2	5
correct rate	0/20	0/20	0/20	0/20	0/20	0/20	0/20	20/20

Table 2.7: The optimal number of clusters of FBSA for DataSet2

Tables 2.6, 2.7 and 2.8 give the results for the number of clusters all validity indices are applied to **FBSA** for the three data sets. Because the results are the same for each run, we list the first and the last values. For DataSet1 and DataSet2, applying V_{WSJ} to **FBSA**, we get the optimal number of clusters is the true number of clusters in all 20 runs. However, applying the existing validity indices to **FBSA**, none obtain the true cluster number. For DataSet3, V_{Xie}, V_{FS}, V_{RH} and V_{WSJ} lead to the correct result, 10 clusters. The other validity indices fail to produce the correct number.

2.4.2.2 Error between cluster prototype and component mean

The other criterion we used here was the error between cluster center and component mean. One of the important goals of a clustering algorithm is to find the cluster prototypes that represent the component means. For this reason, and since the component means are known for the data sets used, we use the error between cluster prototype

	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
run1~run20	2	2	10	10	10	2	5	10
correct rate	0/20	0/20	20/20	20/20	2/20	0/20	0/20	20/20

Table 2.8: The optimal number of clusters of FBSA for DataSet3

and component mean as a criterion. The error $E(c_{Opt})$ is defined as follows: suppose $\{v_i\}(i = 1, 2, ..., c_{Opt})$ are the cluster centers and $\{m_j\}(j = 1, 2, ...K)$ the component means (K is the number of components). Then

$$E(c_{Opt}) = \sum_{i=1}^{c_{Opt}} \min_{1 \leq j \leq K}(\|v_i - m_j\|) \tag{2.11}$$

We calculated the average values of $E(c_{Opt})$ yielded by **Algo2** and **FBSA** with different validity indices for the three data sets over 20 runs. Table 2.9 lists the results for **Algo2** and Table 2.10 lists the results for **FBSA**. According to the explanations of the experimental setting given at the beginning of Section 5.2, if two validity indices always yield the same optimal number of clusters, then they always yield the same clusters. This explains why several validity indices yield the same average value of $E(c_{Opt})$ for a data set.

From the two tables, we note that the results for **FBSA** and **Algo2** are quite similar. **FBSA** is slightly better than **Algo2** for DataSet3 and slightly worse for DataSet1. Both algorithms perform very well when the validity index V_{WSJ} is used. At least for these three data sets, **FBSA** combined with V_{WSJ} seems to be the best combination. The reason for the good performance displayed by both algorithms when combined with V_{WSJ} is the accurate estimation of the number of clusters. This corroborates the results of the previous subsection. The most important conclusion suggested by these experiments is that the accuracy of the new algorithm **FBSA** does not suffer from the restricted initialization scheme, while it is much more time-efficient, as we will show in section 2.4.4.

Data	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
DataSet1	0.8016	0.8016	0.8016	2.6661	2.9495	0.8016	0.6732	0.4721
DataSet2	0.5738	0.5738	0.5738	0.5447	0.5528	0.5738	0.5738	0.3990
DataSet3	2.1378	2.1378	1.6892	1.9969	9.1726	2.1378	5.0926	1.6892

Table 2.9: The error between cluster prototype and component mean for Algo2, for 3 data sets

Data	V_{PC}	V_{PE}	V_{Xie}	V_{FS}	V_{RH}	V_{ZLE}	V_{RLR}	V_{WSJ}
DataSet1	0.8019	0.8019	0.8019	3.2158	3.9313	0.8014	0.8019	0.3853
DataSet2	0.5738	0.5738	0.5738	0.4567	0.5738	0.5738	0.5738	0.2313
DataSet3	2.1356	2.1356	1.6355	1.6355	1.6355	2.1356	5.2356	1.6355

Table 2.10: The error between cluster prototype and component mean for FBSA, for 3 data sets

2.4.3 Stability Across Different Runs

An interesting property of **FBSA** is its stability across different runs, which can be observed from the tests on the three data sets. The output of these experiments (Tables 2.6-2.8) is independent of the initial cluster centers, whereas when **Algo2** is applied to each of these data sets, the output varies quite significantly depending on the validity function used (Tables 2.3-2.5). The way in which new cluster centers are initialized at each phase of **FBSA** is certainly the reason for its stability. Since **FBSA** performs at least as well as **Algo2** in computing the number of clusters and the cluster centers, this property of stability could make it a more interesting choice than **Algo2** in many practical applications.

2.4.4 Comparison of FBSA to Algo2 in Terms of Time Cost

Here we will show the performance of the new algorithms by comparing their numbers of iterations and the real run time needed for convergence. **Algo2** and **FBSA** both

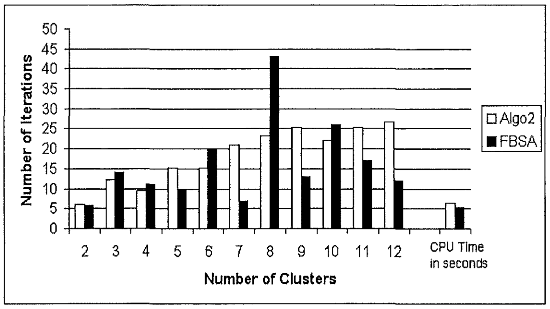


Figure 2.5: Comparison of the number of iterations and running time on DataSet1 for **Algo2** and **FBSA**

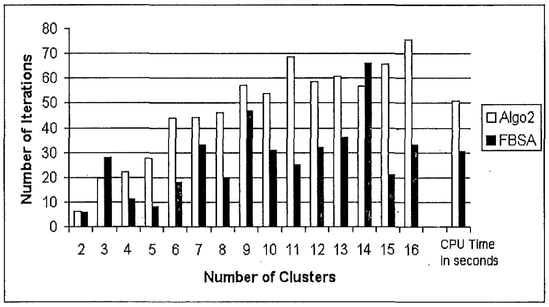


Figure 2.6: Comparison of the number of iterations and running time on DataSet2 for **Algo2** and **FBSA**

use the same validity index, V_{WSJ}. In comparing **FBSA** with **Algo2**, we are interested in both the reduction in the number of iterations for each subsequent value of c tested and the run time. We ran this experiment on a PC with Pentium III 450MHz CPU and 320MB RAM.

2.4.4.1 Test on DataSet1, DataSet2 and DataSet3

Figures 2.5, 2.6 and 2.7 show the numbers of iterations that **Algo2** and **FBSA** need for each value of c and their run times. In all cases, the new algorithm requires less

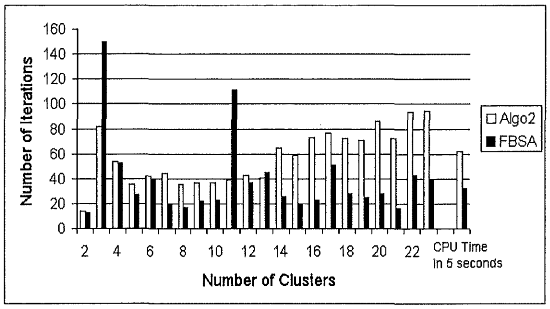


Figure 2.7: Comparison of the number of iterations and running time on DataSet3 for **Algo2** and **FBSA**

running time to converge than **Algo2**. The improvement in the running times is clear. Although there are some variations in the number of iterations for some c, we can see that the reduction in the number of iterations relative to **Algo2** is significant for all data sets, especially when c is large.

2.4.4.2 Test on DataSet4

In addition to the data sets described above, we also tested a larger data set with 60,000 data vectors in order to get a better idea about the improvement of **FBSA** in speed. For statistical purposes, we ran each algorithm 10 times and recorded the average number of iterations for each c and the CPU time. We ran this experiment on a PC computer with a 2.40GHz CPU and 1Gb RAM.

DataSet4 originated from a file provided by Statistics Canada under its Data Liberalization Initiative Program (http://www.lib.unb.ca/gddm/data/ Ftp_famex.html). The FAMEX file from StatsCan is a survey on household expenditures and budgets for the

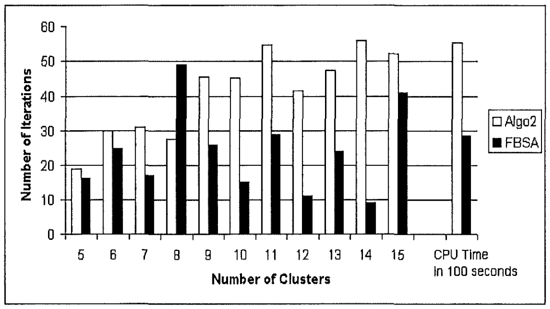


Figure 2.8: Comparison of the number of iterations and run time on DataSet4 for **Algo2** and **FBSA**

year 1996. It includes expenditures, income, and changes in assets and debts. The variables include composition of household, characteristics of dwelling, shelter expenses, food and alcohol, clothing, medical and health care, travel and transportation, recreation and education and tobacco. After a simple preprocessing (removing non-numeric variables and items with missing values), we obtained a data set with 22 variables and 600,000 data vectors. DataSet4 is a randomly selected subset of 60,000 items.

The main objective of testing DataSet4 was to further illustrate the computational efficiency of the proposed algorithm. For a real data set, the actual number of clusters is often unknown. Subjective evaluation of the clustering results is beyond the scope of this thesis. We will restrict ourselves to the evaluation of time efficiency. Figure 2.8 shows the number of FCM iterations needed as a function of c, the number of clusters tested. In the same figure, we also show the comparison of the real CPU time. For the number of iterations, we obtained a similar profile as in the experiments conducted on the three previous data sets. Also, in terms of gain in CPU time, **FBSA** is 48.15% faster than **Algo2**, which is similar to the gain obtained in the previous examples. Thus **FBSA** seems to scale well to a large data set.

2.5 Discussion

From these results and from the results we obtained on similar data sets, we can draw the following conclusions:

1. The new validity index proposed here significantly improves the performance in determining the number of clusters and the accuracy of cluster centers. In fact, based on the three sets of experimental results, only our new validity index is able to yield the correct number of clusters consistently, whatever **Algo2** and **FBSA** is used. The comparison of the cluster centers and the component means shows that **FBSA** combined with V_{WSJ} gives accurate prototypes.

2. The cluster centers yielded by **FBSA** are accurate for small data sets containing fewer than several hundred data (which is the case for most public domain data sets), even when there are overlapping clusters. We also obtained similar results on large data sets generated from a mixture of Gaussians. However, it is difficult to carry out these comparisons on large real-world data sets because the information required for the comparison, such as the true number of clusters and the true cluster centers, is often not available.

3. In terms of computational efficiency, **FBSA** generally needs less run time than **Algo2**. For some c, however, the number of iterations for **FBSA** is significantly larger than for **Algo2.** This situation corresponds usually to the cases where the algorithm is forced to partition a data set into more clusters than it really has.

4. It is easy to understand that the number of iterations for each value of c tends to be small when c is close to the true number of clusters. When the value of c moves

away from the true number of clusters, the algorithm **FBSA** tends to exhibit a more abrupt increase in the number of iterations. Another interesting property of **FBSA** is that the number of iterations for different runs is very stable for each value of c. This is illustrated in Tables 2.6, 2.7 and 2.8. These properties can be useful for developing a strategy to further limit the search range of c.

2.6 Conclusion

The major contributions of this chapter are: 1) a new index for validating clustering results; 2) an improved FCM-based algorithm for determining the number of clusters. The new validity index, V_{WSJ}, is a function of the original data, cluster centers and membership. Experimental results have shown that the new index is able to yield accurate numbers of clusters even for data sets with overlapping clusters, where existing indices often display unpredictable behavior. The new improved clustering algorithm has shown advantages in terms of computation time and stability, compared with the basic trial-and-error FCM-based algorithms. All experiments show that the combination of **FBSA** and V_{WSJ} gets the best results.

Chapter 3

A Theory on Distinguishing Overlapping Components in Mixture Models

In this chapter, we study the overlapping phenomenon in the case of the Gaussian mixture, a fundamental data distribution model for many clustering algorithms. We introduce a novel concept of the ridge curve and establish a theory on the degree of overlap between two components. Based on this theory, we develop an algorithm for calculating the overlap rate. As an example, we use this algorithm to calculate the overlap rates between the classes in the IRIS data set and clarify some of the confusion as to the true number of classes in the data set. We investigate factors that affect the value of the overlap rate, and show how the theory can be used to generate truthed data as well as to measure the overlap rate of a given data set.

3.1 Introduction

The Gaussian mixture model plays an important role in cluster analysis. Many widely used clustering algorithms such as the K-means and the Fuzzy C-means [27] are the most suit on the Gaussian mixture model. These clustering algorithms are widely used in various applications [50, 69, 70, 71, 72, 73]. Their performance often depends on whether the data set contains well separated clusters, or in other words, whether and how the components of the underlined Gaussian mixture overlap to each other. The results of clustering are evaluated on some objective criteria. Examples of such criteria are those based on information theory such AIC, MML and MDL [74][75] and those based on the explicit trade-off between within-cluster variances and between-cluster variances, such as the Xie, Fukuyama and Fisher's discriminant [62][61][76]. Due to the lack of understanding of component overlapping in a mixture, a few researchers evaluate their algorithms with respect to the very fundamental hypothesis, the mixture of Gaussians, based on which the clustering algorithms are designed.

Given a data set satisfying the distribution of a mixture of Gaussians, the degree of overlap between components affects the number of clusters "perceived" by a human operator or detected by a clustering algorithm. In other words, there may be a significant difference between intuitively defined clusters and the true clusters corresponding to the components in the mixture. The component overlapping phenomenon is illustrated in Figure 3.1. Fig. 3.1-a to 3.1-c, show the 1-D case, with two components that are (almost) non-overlapping, partially overlapping and totally overlapping. Fig.3.1-d to 3.1-f show their counterparts in the 2-D case. Clearly, when two components are (almost) non-overlapping or partially overlapping, we expect a clustering algorithm to be able to find

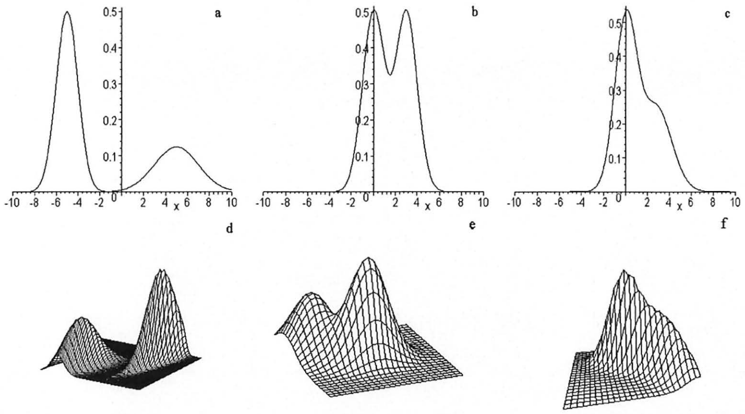


Figure 3.1: Two components that are non-, partially and totally overlapping, in the 1-D and 2-D cases

the two components. On the other hand, when the two components overlap totally, we usually do not expect a basic partition-based clustering algorithm to distinguish between them. It is thus necessary to be able to precisely characterize the degree of overlap in order to measure the "difficulty" of a given set or to generate test sets with a given degree of overlap (difficulty) so that the performance evaluation of a clustering algorithm can be performed on a solid mathematical basis.

Cluster overlap impacts on another important issue of clustering, which is determining the number of clusters. Existing methods, often based on measuring the validity of each possible number of clusters, yield good estimates when the clusters are well separated (non-overlapping). However, for data sets with overlapping clusters, the results are often unpredictable. Examples of such results are reported in our recent paper on determining the number of clusters using Fuzzy C-means based algorithm [77]. One of the main reasons for this problem is that many algorithms fail to distinguish between partially overlapped clusters.

There are some related work to overlapping phenomenon. These work can be cataloged to statistic approach and geometric approach. In statistic approach, which considers the statistic property of the data set, classification error rate is an close concept to overlapping phenomenon [78]. It indicates the probability that an object is assigned to wrong cluster. When cluster strong or total overlapping, it inefficacy. Some distances, for example, Bhattacharyya and Mahalanobis distance, are the ramifications of classification error rate [78, 79]. In geometric approach, which consider the geometric property of the probability density function of the mixture model. Qu and Feng [57] has conducted simulations in order to reveal the relationship between overlapping components and the distance between two adjacent components, for Normal, Poisson and Bi-normal distributions. In a previous study carried out by members of our research group, Aitnouri *et al.*. [58] gave a formal definition of the overlap rate and proposed algorithms for generating data sets with controlled overlap rate. All of these studies apply to the univariate case. In particular, the overlap rate defined in [58] relies on a concept of "apparent width" that does not have an obvious counterpart in the multivariate case. Indeed due to the covariance structure of multivariate components, the phenomenon of component overlapping becomes more complex and need to be investigated in a more general way. We are interested in establishing analytically the relationship between the degree of overlap and the parameters of each component. This relationship will have an important impact in many real applications. For example, generating truthed data sets with prescribed degree of overlap between clusters provides a way of evaluating the capacity of existing clustering algorithms to identify overlapping clusters. In image processing, the degree of overlap between two objects (clusters) in a color image can be used to measure the similarity of the objects [80]. The scale of the overlap rate may hance provide an important argument for merging the two objects.

The main contribution of this chapter is developing a theoretical framework for component overlap in a mixture and a practical algorithm for measuring the degree of overlap between two components. We focus on two components in the 2-D data space. Our investigation is limited to two components for several reasons. First, in most practical cases, one can only interpret the meaning of overlap precisely between two components. For a multi-component mixture, one would need to measure the degree of overlap between one (any) pair of components. Second, establishing the mathematical framework for two overlapped components is a necessary step towards a framework that can account for more than two overlapped components. Finally, determining the components of a mixture containing more than two overlapped components would most likely involve the decidability problem. Since one of the main objectives of this study is to generate truthed data sets with overlapped components, the two-component framework is already a significant leap forward compared to all the current practices, because of the analytical relationship between the overlap rate and the component parameter values [78]. We do not make any special hypothesis regarding the covariance structure. The theory is introduced in the 2-D case and its extension to the multidimensional case will be discussed. The theory is based on a novel concept, that of the "ridge curve". A series of theorems will be established to show the main characteristics of component overlapping. This allows us to give a feasible definition of the overlap rate, as well as developing algorithms for measuring it and generating truthed data sets.

This chapter is organized as follows. In the next section, we prove a series of theorems to establish the theoretical framework to describing the phenomenon of overlapping components. In Section 3, we define the overlap rate between components in a mixture and design a algorithm for calculating this overlap rate. After that, we investigate the IRIS data set. In Section 4, we consider the factors affecting the value of the overlap rate in order to generate truthed data sets with prescribed overlap rates. We conclude our report with a discussion of some extensions of the research.

3.2 Theoretical Framework

In this section, we establish a theoretical framework for describing the overlapping phenomenon between two components in a Gaussian mixture model. The framework is based on the proofs of a series of theorems, which depict the properties of ridge curve.

3.2.1 Mixture Models

Mixture models satisfy some intuitive definitions of cluster structure. Two key properties of a cluster are internal cohesion, which requires that entities within the same cluster should be similar to each other, and external isolation, which requires that entities in one cluster should be separated from entities in another cluster by fairly empty areas of space [81]. Internal cohesion is an inherent property of mixture models, since for a given cluster, data are generated from the same distribution. External isolation concerns the degree of overlap between the components of the mixture model [58].

A set of n entities forming a k-mode two-way array can be presented as $\mathbf{X} = \{X_1, ..., X_n\}$, where X_i ia a vector of dimension d. In the finite mixture models dealt with here, each X_i can be viewed as arising from a mixture of k Gaussian distributions and the probabilistic density function (*pdf*) is given (in the d-dimensional data space) by:

$$\begin{cases} p(X) = \sum\limits_{i=1}^{k} \alpha_i G_i(X, \mu_i, \Sigma_i) \\ X = (x_1, x_2, ..., x_d)^T \in R^d \end{cases} \tag{3.1}$$

with the restrictions $\alpha_i > 0$ for $i = 1, ..., k$ and $\sum\limits_{i=1}^{k} \alpha_i = 1$. (μ_i, Σ_i) denote respectively the mean and the covariance matrix for the i^{th} distribution G_i. G_i is the i^{th} component, given by:

$$G_i(X, \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp{(-\frac{1}{2}(X - \mu_i)^T \Sigma_i^{-1}(X - \mu_i))} \tag{3.2}$$

To study the overlapping phenomenon, we consider the case in which $k = 2$. If the two components are (almost) non-overlapping or only partially overlapping, then the *pdf* of the mixture has two divided local peaks. In this case, the data arising from the mixture model is viewed as two clusters (see Fig.3.1 a, b, d, e). On the other hand, if the *pdf* has only one peak, then it is viewed as two completely (totally) overlapping components. In other words, the two clusters of the data set merge to form one cluster (see Fig.3.1-c and 3.1-f). In what follows, we try to characterize this phenomenon as a function of the parameters of the mixture. In particular, we derive an efficient procedure for verifying whether the two components completely overlap and to compute an overlap rate when they partially overlap.

In what follows, for simplicity reason, we limited our discussion to $d = 2$. In higher dimensional case ($d > 2$), the results can be obtained based on the same argument. In Section 3.3.3, we list the relative formulas in higher dimension.

3.2.2 Ridge Curve and Peaks of the *pdf*

In this subsection, we will prove that the peaks of the *pdf* in R^2 can be found by a search procedure that follows a curve linking the centers of the two components.

As we know, the peaks of the *pdf* satisfy the following system of stationary equations:

$$\begin{cases} \frac{\partial p}{\partial x_1} = A_{x_1}\alpha_1 G_1 + B_{x_1}\alpha_2 G_2 = 0 & (I) \\ \frac{\partial p}{\partial x_2} = A_{x_2}\alpha_1 G_1 + B_{x_2}\alpha_2 G_2 = 0 & (II) \end{cases} \tag{3.3}$$

where

$$\begin{aligned} \begin{pmatrix} A_{x_1} \\ A_{x_2} \end{pmatrix} &= \begin{pmatrix} \frac{\partial}{\partial x_1}(-\frac{1}{2}(X-\mu_1)^T\Sigma_1^{-1}(X-\mu_1)) \\ \frac{\partial}{\partial x_2}(-\frac{1}{2}(X-\mu_1)^T\Sigma_1^{-1}(X-\mu_1)) \end{pmatrix} = -\tfrac{1}{2}\nabla\|X-\mu_1\|^2_{\Sigma_1^{-1}} = -\Sigma_1^{-1}(X-\mu_1) \\ \begin{pmatrix} B_{x_1} \\ B_{x_2} \end{pmatrix} &= \begin{pmatrix} \frac{\partial}{\partial x_1}(-\frac{1}{2}(X-\mu_2)^T\Sigma_2^{-1}(X-\mu_2)) \\ \frac{\partial}{\partial x_2}(-\frac{1}{2}(X-\mu_2)^T\Sigma_2^{-1}(X-\mu_2)) \end{pmatrix} = -\tfrac{1}{2}\nabla\|X-\mu_2\|^2_{\Sigma_2^{-1}} = -\Sigma_2^{-1}(X-\mu_2) \end{aligned} \tag{3.4}$$

Because of the involvement of G_1 and G_2 in Eq.3.3, this system does not have a closed-form solution. A naive numerical solution would imply scanning a region of R^2 directly. The following theorems illustrate that the scanning procedure can be restricted to a curve.

Theorem 1. A general mixture model of two Gaussian distributions, given by Eq.3.1, can be converted to a special form by implementing an affine transformation to X. The special form is given by:

$$\begin{cases} \mu_1 = (\mu_1^1, \mu_1^2)^T = (0,0)^T, \mu_2 = (\mu_2^1, \mu_2^2)^T \\ \Sigma_1 = \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \text{ and } \Sigma_2 = \begin{pmatrix} \eta_1^2 & 0 \\ 0 & \eta_2^2 \end{pmatrix} \end{cases} \tag{3.5}$$

In other words, the two covariance matrices are diagonal and one component is centered at the origin. The proof of this theorem is based on simultaneous diagonalization [82].

Proof of Theorem 1: Based on Cholesky decomposition for symmetric positive definite matrix, Σ_1^{-1} can be decomposed to $\Sigma_1^{-1} = U_1 U_1^T$, where U_1 is a non-singular matrix, and $Y = U_1(X - \mu_1)$. Then Eq.3.1 can be written as:

$$p(Y) = A_1 \exp(-\frac{1}{2}Y^T Y) + A_2 \exp(-\frac{1}{2}(Y - \mu_Y)^T \Sigma_Y^{-1}(Y - \mu_Y)) \tag{3.6}$$

where A_1, B_1 are constants, $\mu_Y = U_1(\mu_2 - \mu_1)$ and $\Sigma_Y = U_1 \Sigma_2 U_1^T$. Based on singular value decomposition for symmetric positive definite matrix, Σ_Y^{-1} can be decomposed to $\Sigma_Y^{-1} = U^T D U$, where D is a diagonal matrix and U is a normal orthogonal matrix $(U^T U = I)$, and $Z = UY$. Then the *pdf* of Z can be written as:

$$p(Z) = A_1 \exp(-\frac{1}{2}Z^T Z) + A_2 \exp(-\frac{1}{2}(Z - \mu_Z)^T D(Z - \mu_Z)) \tag{3.7}$$

where $\mu_Z = U\mu_Y$. This form is the special case declared. $\diamondsuit$

Inference 1. A general mixture model of two Gaussian distributions, given by Eq.3.1, can be converted to a special form by implementing an affine transformation to X. The special form is given by:

$$\begin{cases} \mu_1 = (\mu_1^1, \mu_1^2)^T = (0,0)^T, \mu_2 = (\mu_2^1, 0)^T \\ \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \Sigma_2 = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} \\ \sigma_{12} & \sigma_{22}^2 \end{pmatrix} \end{cases} \tag{3.8}$$

Now we introduce the important concept of the ridge curve (RC) of a mixture model of two Gaussian distributions as follows.

Definition 1. Given a mixture model of two Gaussians (Eq.3.1), the quadratic curve:

$$A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0 \tag{3.9}$$

is called the ridge curve (RC) of the mixture model. Here, A_{x_1}, B_{x_2}, B_{x_1} and A_{x_2} are defined by Eq.3.4.

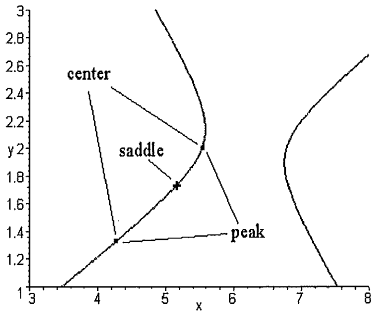


Figure 3.2: The ridge curve of the mixture in which the parameters given by Eq.3.20. Here, the peaks of *pdf* and the means of the mixture are almost superposition.

Fig.3.2 shows the ridge curve of the Gaussian mixture defined by following parameters:

$$\begin{cases} \alpha_1 = 0.5, \alpha_2 = 0.5, \mu_1 = \begin{pmatrix} 4.26 \\ 1.33 \end{pmatrix}, \mu_2 = \begin{pmatrix} 5.55 \\ 2.03 \end{pmatrix} \\ \Sigma_1 = \begin{pmatrix} 0.22 & 0.07 \\ 0.07 & 0.04 \end{pmatrix} \text{ and } \Sigma_2 = \begin{pmatrix} 0.29 & 0.05 \\ 0.05 & 0.07 \end{pmatrix}. \end{cases} \tag{3.10}$$

The mixture comes from IRIS data set. Feature 3 and 4 are selected to be 2 dimensions and class 2 and 3 are two components. In the figure, the mean of two components and the points with the peaks of *pdf* are superposition. The point with the saddle of the *pdf* is on the middle of the two means.

Theorem 2. The means of the two components and the stationary points (peak points and saddle points) of the *pdf*, $p(X, \mu_1, \mu_2, \Sigma_1, \Sigma_2, \alpha_1, \alpha_2)$, are on the ridge curve.

Proof: The first part of this theorem can be easily proved by considering that both $A_{x_1}B_{x_2}$ and $B_{x_1}A_{x_2}$ contain the factors $(X - \mu_1)$ and $(X - \mu_2)$.

For the second part, we mentioned above that any stationary point (x_1, x_2) of the *pdf* should satisfy Eq. 3.3. The stationary equation can be written as:

$$\begin{pmatrix} A_{x_1} & B_{x_1} \\ A_{x_2} & B_{x_2} \end{pmatrix} \begin{pmatrix} \alpha_1 G_1(X, \mu_1, \Sigma_1) \\ \alpha_2 G_2(X, \mu_2, \Sigma_2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.11}$$

We know that, for any $X = (x_1, x_2)$, the vector $\begin{pmatrix} \alpha_1 G_1(X, \mu_1, \Sigma_1) \\ \alpha_2 G_2(X, \mu_2, \Sigma_2) \end{pmatrix}$ is non-zero. Thus the matrix $\begin{pmatrix} A_{x_1} & B_{x_1} \\ A_{x_2} & B_{x_2} \end{pmatrix}$ is singular. This means $\begin{vmatrix} A_{x_1} & B_{x_1} \\ A_{x_2} & B_{x_2} \end{vmatrix} = 0$ or $A_{x_1} B_{x_2} - B_{x_1} A_{x_2} = 0$. Thus the stationary points of the *pdf* are on the RC. $\diamondsuit$

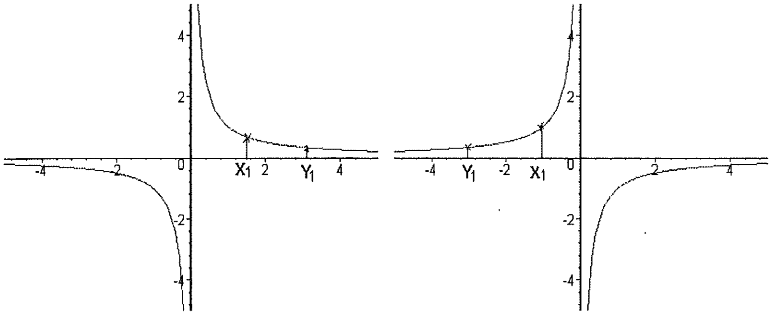
Based on **Theorem 2**, we conclude that it is sufficient to search the ridge curve to find the stationary points of the *pdf*. However, due to the quadratic nature of the curve, the search procedure must solve the double-choice problem, in addition to restraining the search interval. Indeed, the following theorem will indicate that all of the stationary points of the *pdf* and the means of the components fall on the same segment of the ridge curve.

Theorem 3. The stationary points of the *pdf* described by Eq. 3.1 fall on the segment between the two means of the components of the ridge curve.

For the proof of this theorem, we need to introduce some lemmas necessary for the special case mentioned in **Theorem 1**.

We notice the following fact: let $xy = a$ denote a hyperbola in R^2, and (x_1, x_2) and (y_1, y_2) be two points on the hyperbola. The two points are on the same branch iff $x_1 y_1 > 0$ or $x_2 y_2 > 0$. See Fig. 3.3

Lemma 1. Based on the special case mentioned in **Theorem 1**, the ridge curve, $A_{x_1} B_{x_2} - B_{x_1} A_{x_2} = 0$, is a hyperbola or a line.



$a > 0$ $a < 0$

Figure 3.3: The two points are on the same branch iff $x_1y_1 > 0$ or $x_2y_2 > 0$

Proof: Let $b = \frac{1}{\sigma_1^2\eta_2^2} > 0$ and $c = -\frac{1}{\sigma_2^2\eta_1^2} < 0$. The curve $A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0$ can be written as $(b+c)x_1x_2 - b\mu_2^2x_1 - c\mu_2^1x_2 = 0$.

If $b+c=0$ or if $\mu_2^1\mu_2^2 = 0$, then the above equation degenerates to a line. If $b+c \neq 0$ and $\mu_2^1\mu_2^2 \neq 0$, implementing the following affine transformation to $X = (x_1, x_2)$,

$$\begin{cases} x_1^{'} = x_1 - \frac{c\mu_2^1}{b+c} \\ x_2^{'} = (b+c)x_2 - b\mu_2^2 b + c \end{cases} \tag{3.12}$$

we get the normalized hyperbolic curve: $x_1^{'}x_2^{'} = \frac{bc\mu_1^2\mu_2^2}{b+c}$. $\Diamond$

Lemma 2. Under the same conditions as for **Lemma 1**, the stationary points of Eq.3.1 are inside the rectangle defined by following two diagonal points: $((0,0),(\mu_2^1,\mu_2^2))$.

Proof: Suppose that a stationary point $X_0 = (x_1^0, x_2^0)$ is outside the rectangle. Without loss of generality, we suppose that $\mu_2^1 > 0$ and X_0 is to the right of the rectangle, i.e. $x_1^0 > \mu_2^1$. Consider the radial:

$$\begin{cases} x_1 = \mu_2^1 + t(t > 0) \\ x_2 = x_2^0 \end{cases} \tag{3.13}$$

Both $(X-\mu_1)^T\Sigma_1^{-1}(X-\mu_1)) = X^T\Sigma_1^{-1}X$ and $(X-\mu_2)^T\Sigma_2^{-1}(X-\mu_2))$ are monotonically increasing. Thus the value of $p(X)$ is monotonically decreasing on the radial. This means that X_0 is not a stationary point of the *pdf*. $\diamondsuit$

Combining **Theorem 2** and **Lemmas 1** and **2**, we obtain the proof of **Theorem 3** as follows.

Proof of Theorem 3: According to **Theorem 1**, an affine transformation $X = mZ+N$ (m is a 2×2 affine matrix and N is 2-D vector), $Z = (z_1, z_2) \in R^2$, transforms the mixture to the special case mentioned in **Theorem 1**. The ridge curve $A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0$ is transformed to $A_{z_1}B_{z_2} - B_{z_1}A_{z_2} = 0$ and the stationary points of *pdf* about X are translated to the stationary points about Z, because the transform is linear. Based on **Lemmas 1** and **2**, the stationary points of the *pdf* about Z fall on the segment between the two means of the components of the curve, $A_{z_1}B_{z_2} - B_{z_1}A_{z_2} = 0$. Thus the stationary points of the *pdf* described by Eq. 3.1 fall on the segment between the two means of the components of the curve, $A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0$. $\diamondsuit$

Based on the result of **Theorem 3**, for determining the stationary points of Eq.3.1, we need only search the segment of the *RC*, $A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0$, between the two means. So that the procedure of searching the stationary points of a Gaussian mixture is restricted on a linear searching procedure.

3.3 Degree of Overlap

In this section, we will give the definition of the *overlap rate* between two components of a mixture, design an algorithm to compute it and test the new concept on the IRIS data set.

3.3.1 Definition of the Overlap Rate

In general, a definition for the overlap rate implements the following principle: 1) the overlap rate tends to decrease (→ 0) as the two components become more separated, 2)the overlap rate increases (→ 1) as the two components become more strongly overlapped. An intuitive definition of the overlap rate was given in [58] as a ratio of the minimum between the two component center (if such a minimum exists) and the lower maximum also situated between the component centers. This definition does not have a natural extension to the multi-dimensional case since there is no local minimum between the two component centers. The following definition relies on the ridge curve concept developed in the previous section.

Definition 2. The overlap rate of two Gaussian components in a mixture is defined by:

$$OLR(G_1, G_2) = \begin{cases} 1 & \text{if } p(X) \text{ has one peak in } RC \\ \frac{p(X_{Local\_Min})}{p(X_{Sub\_Max})} & \text{if } p(X) \text{ has two peaks in } RC \end{cases} \tag{3.14}$$

where $X_{Local\_Min} = \arg(Local\_Min_{X \in C}\ p(X))$ is the local minimum point of $p(X)$ on the ridge curve and $X_{Sub\_Max} = \arg(Sub\_Max_{X \in C}\ p(X))$ is the lower peak point of $p(X)$ on the ridge curve. It is not difficult to show that this satisfies the intuitive principle for the overlap rate.

The value of OLR describes the level of overlapping between two components (clusters). It does not the value of percent of data points locating the "overlapping region". And it is not linear to the value of the percent. Depending our experiments, if the value of OLR is less than 0.6 then the two components (clusters) can believed well separated, if the OLR belongs to (0.6, 0.8] then they are partial overlapping and the OLR is larger than 0.8 then they are strong overlapping (see Fig 3.10-3.14.

3.3.2 Algorithm for Calculating OLR

To compute OLR, the stationary points of $p(X)$ need to be determined. Indeed, considering the special case of the conditions in **Theorem 1**, these points are the solution of the following equations:

$$\begin{cases} A_{x_1}\alpha_1 G_1 + B_{x_1}\alpha_2 G_2 = 0 & (I) \\ A_{x_1}B_{x_2} + A_{x_2}B_{x_1} = 0 & (II) \end{cases} \tag{3.15}$$

Expressing x_2 in terms of x_1 from (II) and substituting x_2 in (I), we obtain a complex equation: $M_1(x_1)e^{N_1(x_1)} + M_2(x_1)e^{N_2(x_1)} = 0$, where $M_1(x_1), M_2(x_1), N_1(x_1)$ and $N_2(x_1)$ are rational functions of x_1. This equation does not have a closed-form solution. For this reason, we have designed a numerical algorithm for calculating the stationary points of $p(X)$ based on **Theorem 3**. The main idea of the algorithm is to search the segment of the curve $A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0$ between the means of the two components. A local maximum point is a peak of the *pdf*, and the minimum point is a saddle point. Algorithm **COLR** below computes the overlap rate of two mixture components. Using this algorithm, we can estimate the overlap rate of any two clusters in a given set of data by first estimating the mean, covariance matrix and prior probability of each cluster.

Algorithm COLR (for computing OLR of the mixture in Eq.3.1)

1. Input the parameters of two distributions $(\mu_1, \mu_2, \Sigma_1, \Sigma_2, \alpha_1, \alpha_2)$.

2. Compute the ridge curve $A_{x_1}B_{x_2} - B_{x_1}A_{x_2} = 0$.

3. Move from μ_1 to μ_2 on RC, finding the maximum and minimum points of $p(X)$.

4. Compute OLR of the two components by Eq. 3.14.

3.3.3 Extension to high Dimensional Data

The theory on overlap presented previously can naturally be extended to high dimensional case. In real applications, high dimensional data (3 or more than 3 dimensions) are more popular than low dimensional (one or two dimensions) data. In d-dimensional case ($d > 2$), the two main concepts of the theory, ridge curve and overlap rate, can be described as follows.

The stationary points of the *pdf* function 3.1 satisfy

$$\begin{cases} \frac{\partial p}{\partial x_1} = A_1\alpha_1 G_1 + B_1\alpha_2 G_2 = 0 \\ \frac{\partial p}{\partial x_2} = A_2\alpha_1 G_1 + B_2\alpha_2 G_2 = 0 \\ \ldots\ldots \\ \frac{\partial p}{\partial x_d} = A_d\alpha_1 G_1 + B_d\alpha_2 G_2 = 0 \end{cases} \tag{3.16}$$

where, A_{x_i} and B_{x_i} ($i = 1, 2, ...d$) are defined by:

$$\begin{aligned} A_i &= \tfrac{\partial}{\partial x_i}(-\tfrac{1}{2}(X-\mu_1)^T\Sigma_1^{-1}(X-\mu_1)) = -\tfrac{1}{2}(\nabla\|X-\mu_1\|^2_{\Sigma_1^{-1}})_i = (-\Sigma_1^{-1}(X-\mu_1))_i \\ B_i &= \tfrac{\partial}{\partial x_i}(-\tfrac{1}{2}(X-\mu_2)^T\Sigma_2^{-1}(X-\mu_2)) = -\tfrac{1}{2}(\nabla\|X-\mu_2\|^2_{\Sigma_2^{-1}})_i = (-\Sigma_2^{-1}(X-\mu_2))_i \end{aligned} \tag{3.17}$$

The ridge curve of Gaussian mixture 3.2 is defined by following $d-1$ equations:

$$\begin{cases} A_1 B_2 - B_1 A_2 = 0 \\ A_2 B_3 - B_2 A_3 = 0 \\ \ldots\ldots \\ A_{d-1} B_d - B_{d-1} A_d = 0 \end{cases} \tag{3.18}$$

Based on the concept of ridge curve in d-dimensional case (Eq. 3.18), the overlap rate between G_1 and G_2, $OLR(G_1, G_2)$, has the same formula as in the 2-dimensional case. The computation of the OLR can be carried out in the same way to **Algorithm COLR**.

3.3.4 Measuring the Overlap of Clusters in the IRIS Data Set

The aim of this experiment is to show how our algorithm can be used to measure degrees of overlap between user-defined clusters. We have chosen the IRIS data set because it is the most commonly used benchmark set in cluster analysis. The IRIS data set is a biometric data set consisting of 150 measurements belonging to three flower varieties. Each class contains 50 observations, in which four variables, the length and width of both petal and sepal, are measured. The data are represented as points in a 4-dimensional measurement space. Pal and Bezdek [67] suggest that since two of the three classes overlap substantially, one can argue in favor of either 2 or 3 classes. Halgamuge and Glesner [68] have shown that a very good classification can be obtained by using only two features (feature 3 and 4). There are various degrees of overlap on the pairs of variables (features) chosen. For purposes of illustration, we will provide a precise measurement of the overlap rates between different classes as they are projected onto subspaces generated by each pair of feature components. We will also give a precise measurement of the overlap rates between different classes taken in R^4.

We tested all combinations of two features. Fig.3.4 illustrates IRIS data projected onto each two-feature subspace. We used the following Maximum Likelihood formula to estimate the parameters of each component (class):

$$\begin{cases} \mu_i = \frac{1}{n_i}\sum_{j=1}^{n_i} X_j \\ \Sigma_i = \frac{1}{n_i-1}\sum_{j=1}^{n_i}(X_j-\mu_i)^T(X_j-\mu_i) \\ \alpha_i = n_i/n \\ i = 1,2,3 \end{cases} \tag{3.19}$$

Fig.3.5 shows the *pdf* of the mixture models based on Formula 3.19. The overlap

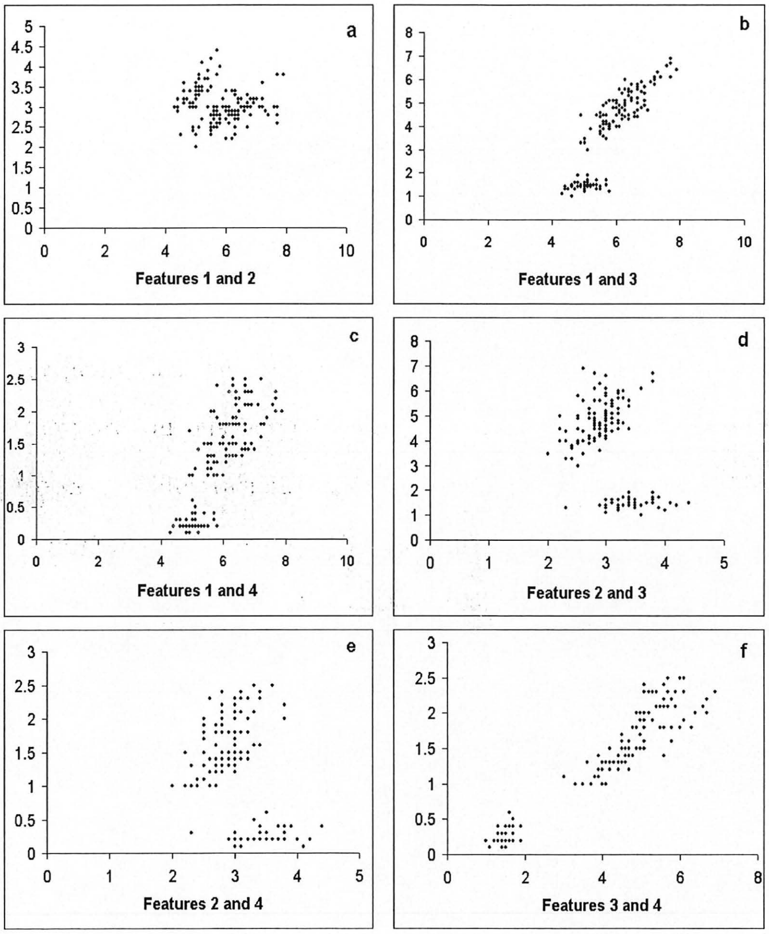


Figure 3.4: Projected IRIS data for each pair of features: a-features (1,2), b-features (1,3), c-features (1,4), d-features (2,3), e-features (2,4) and f-features (3,4)

phenomena shown in Figure 3.4 are naturally illustrated in the figures of the corresponding *pdf*s in Figure 3.5. For instance, classes 2 and 3 strongly overlap if feature 1 and 2 are chosen (Fig.3.4-a). The *pdf*s of these two classes merge to a (large) one-mode distribution in Fig.3.5-a. These two classes exhibit partially overlapped distributions when any other pair of feature is chosen. It should also be mentioned that class 1 is always well separated from the other classes, whichever pair of features is chosen.

Algorithm COLR was used to compute the OLR for each pair of classes. Table 3.1 lists the OLRs of each pair of classes for each group of features. From this table, one can easily see that classes 2 and 3 are quite strongly overlapped for any pair of features. In particular, they are totally overlapped if feature group (1,2) is chosen (the overlap rate between classes 2 and 3 reaches 1). The IRIS data also overlap strongly if group (2,3) is chosen. On the other hand, feature group (2,4) seems to give the best overall between-class separation. Using feature group (3, 4), the OLR of class 2 and 3 is a not high. This accords with the result of [68]. An important asset of **Algorithm COLR** is thus that it can be used to solve the subset feature selection problem, especially if the selection aims to improve classification accuracy.

Table 3.2 shows the results of all of 4 features used. It is interesting to note that the overlap rate for any pair of classes is much lower than when subsets of features are used. In fact, the 3 classes are well separated in the original 4-dimensional space. In the cluster analysis literature, many authors claim, erroneously, that the IRIS data could be considered as a 2-cluster data set as well as a 3-cluster data set, based solely on visual observation of the 2-D projection of the data set. However, any person who has built a Bayes classifier with a Gaussian *pdf* for each class should have noticed that the classifier performs very well with any reasonable partition of the original data set into a training

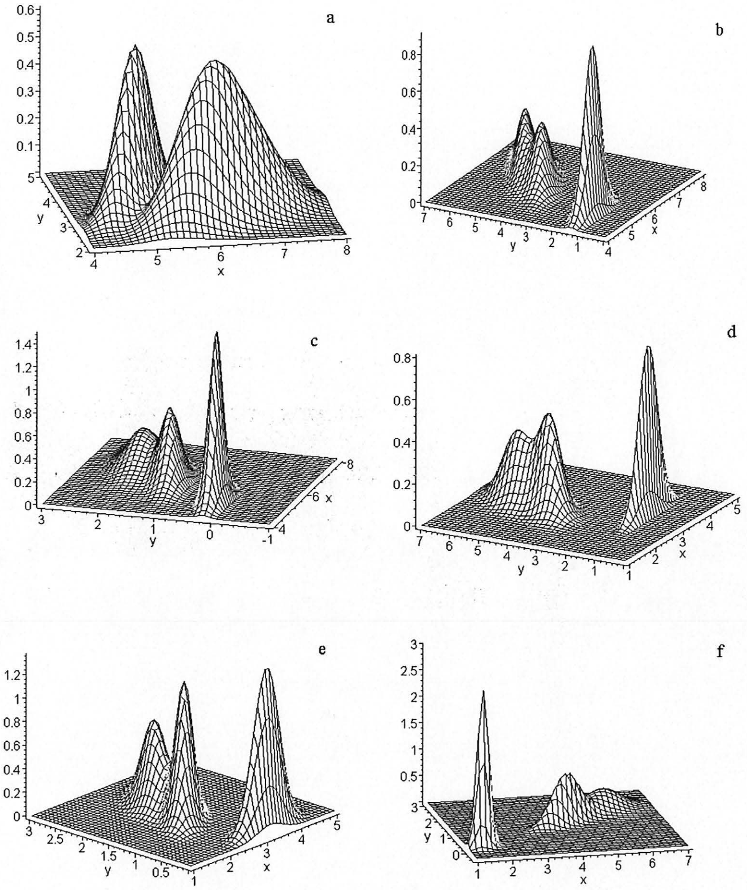


Figure 3.5: *pdf* for each pair of features: a-features (1,2), b-features (1,3), c-features (1,4), d-features (2,3), e-features (2,4) and f-features (3,4)

Features	(1, 2)	(1, 3)	(1, 4)	(2, 3)	(2, 4)	(3, 4)
Class 1, 2	0.115	0.0001	0.004	0.0001	0.0001	0.0001
Class 1, 3	0.111	0	0	0	0	0
Class 2, 3	1	0.683	0.778	0.895	0.567	0.776
Number of Classes	2	3	3	3	3	3

Table 3.1: The OLR for each group of features and each pair of classes

	Class 1, 2	Class 1, 3	Class 2, 3
OLR	3.39×10^{-6}	1.66×10^{-11}	0.524

Table 3.2: The OLR for each pair of classes when all feature are used

set and a test set. The overlap rate concept proposed in this paper allows for a better explanation of these results.

3.4 Generating Truthed Data Sets with prescribed OLRs

In this section, we propose a general framework for generating truthed data sets. We restrict our discussion to the case in which there are two components in a mixture. The mixture with multiple overlapped components can be made up of various separate pairs of components.

3.4.1 The Factors Affecting the OLR

It is difficult to give a closed-form solution to generating parameters of the mixture models with partially overlapped components. The difficulty arises from the complexity of creating an analytic expression to describe the relationship between the degree of overlap and the parameters of the components. In the 1-dimensional case, E. Aitnouri *el* *al* [58] give an approximate solution to the problem based on a linear approximation to the Gaussian components. However, in the multidimensional case, creating an effective approximation to the multivariate Gaussian components and solving an equation like Eq. 3.15 is a hard task. In this section, we try to solve this problem by a numerical method: adjusting one parameter, while keeping the others fixed, to match the prescribed overlap rate between components in a mixture. For this reason, we consider the factors that affect the OLR. The aim of the following subsections is to show the influence of different parameters of the mixture on the OLR. It is very important to understand the relationships between the mixture parameters and the overlap rates if one wants to generate test data with controlled overlap rates.

Based on the discussion in the Section 3.2, we restricted our investigation to the special case defined in **Inference 1**. Let G_1 and G_2 be two Gaussian components of a mixture model. Without loss of generality, we suppose that the parameters of the two components are given by:

$$\begin{cases} G_1 : \alpha_1 = 0.5,\ \mu_1 = (0,0),\ \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ G_2 : \alpha_2 = 0.5,\ \mu_2 = (3,0),\ \Sigma_2 = \begin{pmatrix} 2.1751 & 1.825 \\ 1.825 & 2.1751 \end{pmatrix} \end{cases} \tag{3.20}$$

In what follows, we will show the evolution of the OLR when one of the parameters is varied.

3.4.1.1 Effects of Varying the Mixing Coefficient

Fig.3.6 shows the effects of varying the mixing coefficient $\alpha_1 : 0 \rightarrow 1$. The OLR is a piecewise continual function of α_1. Experimental results show that the OLR reaches

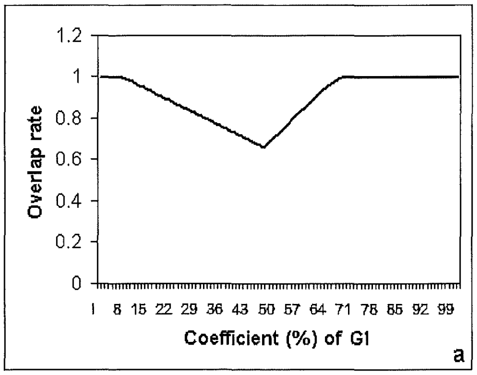


Figure 3.6: The effects of varying the mixing coefficient on the OLR

its minimum (represented by r_{min}) when $\alpha_1 = 0.46$. The fact that the OLR reaches 1 at both ends means that the two components overlap completely when the coefficient crosses a threshold. The difference between the two covariance matrices is the main reason for the asymmetry of the OLR curve. We conclude that, for a given OLR ($\geq r_{min}$ and ≤ 1, we can find a pair of coefficients to match the OLR.

3.4.1.2 Effects of Varying the Distance Between the Two Means

Fig.3.7 shows the relationship between the OLR and the distance between the two means. For this experiment, we varied μ_2 from (0, 0) to (8, 0). When the two means are very close to each other (distance < 2.16), the OLR is 1. The OLR decreases rapidly to zero once it falls below 1 (once two partially overlapped components appear). We notice that modifying the distance between of the two means leads to values of the OLR covering $[0, 1]$. So we conclude that, for a given OLR ($0 \leq OLR \leq 1$), we can find a value of the distance between two means to match the OLR. The distance between the two means is a key parameter to control the overlap rate.

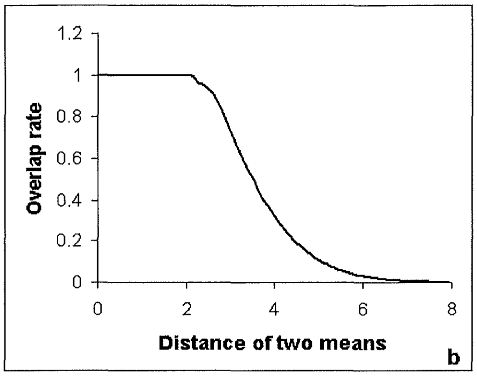


Figure 3.7: The effects of varying the distance between the two means on the *OLR*

3.4.1.3 Effects of Varying the Covariance Matrix

Describing the relationship between the *OLR* and the difference between the two covariance structures is more complex. Under the above simplified assumption, the isohypse of the *pdf* of G_1 is a circle, while the isohypse of the *pdf* of G_2 is an ellipse. The difference between the two covariance structures can be characterized by three factors: The first is the angle between the two main axes of the isohypses (the main axis of the circle is the x-axis). The second is the ratio between the principal and secondary axes of the ellipse. The third is the scale of the main and secondary axes. Fig.3.8 shows the two isohypses, main axis and secondary axis. Because the first isohypse of G_1 is a circle, the angle between the two main axes is equal to the angle between the x-axis and the main axis of G_2's isohypse, θ_2, as shown in Fig.3.8, and only the main and secondary axes of G_2's isopypse are considered. We keep the mixing coefficients and means of the mixture unchanged in the following experiments.

At first, we consider the effects of varying the angle between the main axes of the two isohypses. For this factor (see Fig.3.9-a), the angle θ_2 varies from $-\pi/2$ to $\pi/2$. It

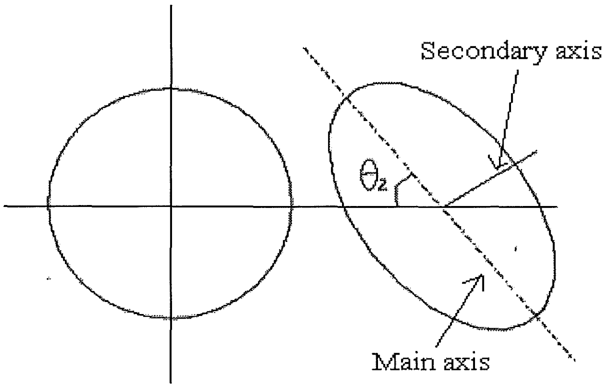
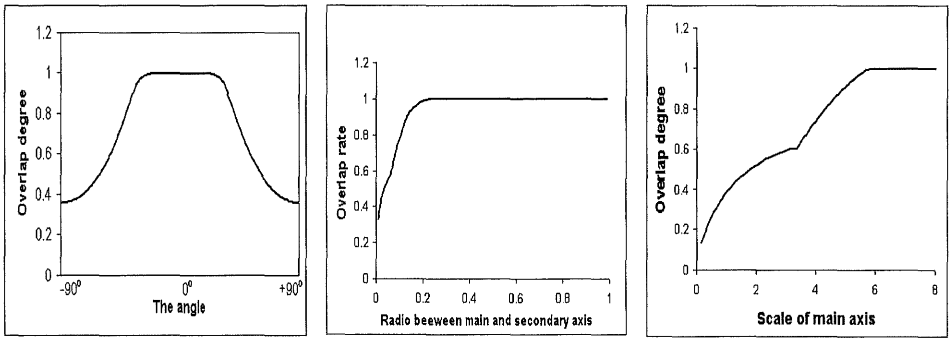


Figure 3.8: Isohypses of the two components



a b c

Figure 3.9: The effects of varying the two covariance matrices on the *OLR*

is easily understood that OLR reaches its maximum (not necessarily 1 and represented by r_{max}) around $\theta_2 = 0$ and reaches its minimum (not necessarily 0 and represented by r_{min}) around $\theta_2 = \pm\pi/2$, since the principal axis of the ellipse is (almost) aligned with the center of the circle (or vertical). If the given OLR is in $[r_{min}, r_{max}]$, we can find out two values (positive or negative) for matching the OLR.

Next, the effects of changing the ratio between the principal and secondary axes are shown in Fig.3.9-b. The curve shows the change in the overlap rate as the secondary axis of one isohypse increases to equal the main axis. We keep the angle $\theta_2 = \pi/4$ and the "main axis" of G_2 =2.0. The "secondary axis" varies from 0.02 to 2.0. The curve shows that the overlap rate starts from its minimum, r_{min}, and increases rapidly to 1.0 as the secondary axis increases. This means we can find a value of the secondary axis of G_2 to match the given OLR ($\geq r_{min}$ and ≤ 1).

Finally, the effects of the scale of the principal and secondary axes has effects corresponding to the second (see Fig.3.9-c). However, in this term, the curve indicates the change in the overlap rate as the "main component" (the component with higher peak) changes to "secondary component" (the component with lower peak). The junction between the two arc segments is the point at which the main component translates to the secondary component. In this curve, the value of the OLR covers $[0, 1]$. Thus we can choose a scale of the principal and secondary axes to match any given OLR ($\in [0, 1]$).

3.4.2 Examples of Generating Truthed Data Sets

In this subsection, we show some truthed data sets generated to match certain overlap rates by modifying the different parameters mentioned above.

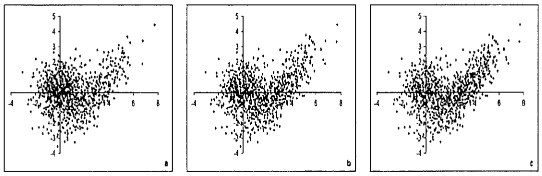


Figure 3.10: Generated data sets based on the coefficients: a) OLR is 0.99, coefficients are 0.67 and 0.33; b) OLR is 0.8, coefficients are 0.54 and 0.46; c) OLR is 0.67, coefficients are 0.46 and 0.54

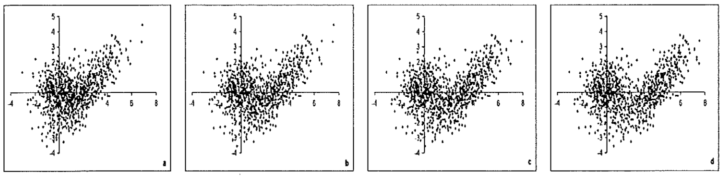


Figure 3.11: Generated data sets based on the distance between the two means: a) OLR is 0.99, distance is 2.16; b) OLR is 0.8, distance is 2.84; c) OLR is 0.60, the distance between two means is 3.28; d) OLR is 0.40, distance is 3.76.

We chose 0.99, 0.80, 0.60 and 0.40 as four prescribed OLRs. The initial components are given by Eq.3.20. The first example involves choosing the coefficients of the mixture described in Section 4.1 to match the certain OLRs. Fig.3.10 shows three data sets. Their OLR are 0.99, 0.80 and 0.67. There are no coefficients that yield the OLRs of 0.6 and 0.4. We replace 0.60 by 0.67 to generate a similar data set.

Fig.3.11, 3.12, 3.13 and 3.14 show the data sets generated based on adjusting each of the other parameters.

3.4.3 Evaluating Objectively Validity Indices

As mentioned in Chapter and 1, determination of the number of clusters is still a open problem in cluster analysis. Many "validity indices" have been proposed for

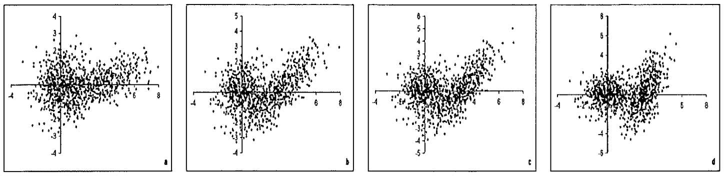


Figure 3.12: Generated data sets based on the angle between two main axes: a) OLR is 0.99, angle is 14.9^o; b) OLR is 0.80, angle is 40.7^o; c) OLR is 0.60, angle is 52.1^o; d) OLR is 0.40, angle is 73.3^o.

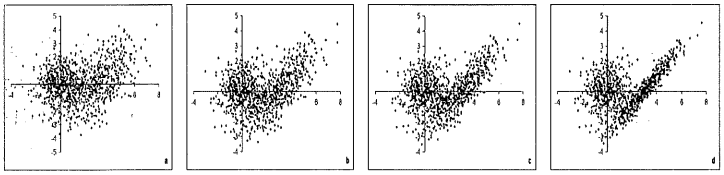


Figure 3.13: Generated data sets based on the ratio between the axes of G_2 : a) OLR is 0.99, ratio is 0.23; b) OLR is 0.80, ratio is 0.115; c) OLR is 0.60, ratio is 0.065; d) OLR is 0.40, ratio is 0.018

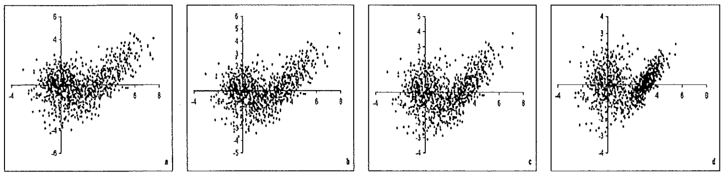


Figure 3.14: Generated data sets based on the scale of the principle axis of G_2: a) OLR is 0.99, axes is 2.4; b) OLR is 0.80, axes is 2.08; c) OLR is 0.60, axes is 1.75; d) OLR is 0.40, scale is 1.06

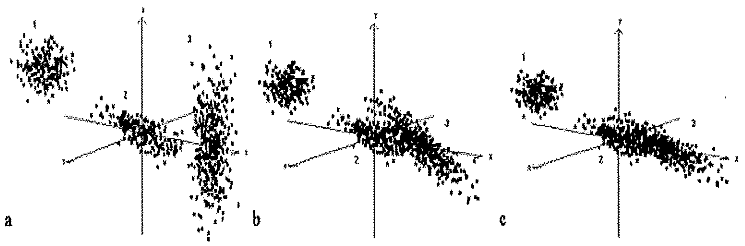


Figure 3.15: Generated data sets : a) OLR is 0.07 in $X_{1,1}$, b) OLR is 0.50 in $X_{1,2}$, and c) OLR is 0.80 in $X_{1,3}$.

this purpose. The performance of a validity index depends, however, on how clusters overlap each other. We are therefore interested in evaluating the ability of existing validity indices to identify overlapping clusters. The theory on overlap between clusters provides a physical measure of the complexity of a data set and lays down a foundation for generating truthed overlapped data sets with prescribed degrees of overlap between clusters, thus making it possible to rate the performance of a validity index.

We generated 5 groups of data sets (each group having three data sets containing three clusters) with controlled OLR, based on the theory discussed above. Each data set is represented by $X_{g,s}$, with g (the number of group) varying from 1 to 5 and s (the number of set) from 1 to 3. Due to space limitations, our discussion focuses on the results for these individual data sets only (instead of full statistics). The first four groups of data sets are similar in that the data sets in each are 3-dimensional, the first cluster is well separated from the two others, and the OLR between cluster 2 and cluster 3 is respectively 0.07, 0.50 and 0.80 for the different sets in each group.

Each set in group X_1 has 800 points. $X_{1,1}$ is a mixture of ellipsoidal and spherical clusters with different number of data points in each cluster (Fig.3.15). The OLR between clusters 2 and 3 in $X_{1,1}$ is 0.07. By varying the angle between the main axis of cluster 3 and the x-axis, we generate the two other data sets $X_{1,2}$, and $X_{1,3}$ with different OLR between clusters 2 and 3 (Fig. 3.15(b) and (c)). For $X_{2,1}$ of the second group, X_2, the first cluster has the form of a sphere with 200 data vectors. The second one is large and has the form of a hyper-ellipsoid with 450 data vectors. The last one is dense and has the form of an ellipsoid with 250 data vectors. $X_{2,2}$ and $X_{2,3}$ are generated based on $X_{2,1}$ by moving the center of cluster 3 closer to the center of cluster 2. The third group, X_3, is very similar to X_2 in the way $X_{3,2}$ and $X_{3,3}$ are generated. The major difference is in the first data set $X_{3,1}$, in which there are only spherical clusters. The numbers of data points in the clusters are respectively 300, 500, and 200. The fourth group, X_4, has 600 data vectors. The first and second clusters of $X_{4,1}$ are spherical, with 150 and 200 data vectors respectively. The last one has the form of an ellipsoid with 250 data vectors. The main axis of the third cluster (the ellipsoid) forms an angle of 30 with the x-axis. $X_{4,2}$ and $X_{4,3}$ are generated by scaling the secondary axis of cluster 3 (while keeping the principal axis fixed). The last group, X_5, contains three 2-dimensional data sets ($X_{5,1}$, $X_{5,2}$, $X_{5,3}$). Each contains two spherical clusters and one ellipsoidal cluster. Each cluster has 500 data vectors. The significant overlap is between one spherical cluster and the ellipsoidal cluster. Contrary to X_4, both axes of the ellipsoidal cluster are changed to obtain the three overlap rates.

Table 3.3 summarizes the results of the main validity indices introduced in Section 2.2 in conjunction with the FCM algorithm for the five groups of data sets.

The results in the preceding table invite several comments.

1. The behavior of V_{PC} in all our experiments is almost the same. It fails even when OLR is low. In general we can say that V_{PC} is very sensitive to overlapping clusters. It performs well only with well-separated clusters.

data sets	OLR	Validity Indices						
		V_{PC}	V_{FS}	V_{Xie}	V_{ZLE}	V_{RH}	V_{RLR}	V_{WSJ}
$X_{1,1}$	0.07	2	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{1,2}$	0.50	2	9	9	9	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{1,3}$	0.80	2	4	2	2	2	4	4
$X_{2,1}$	0.07	$\underline{3}$	4	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{2,2}$	0.50	2	5	$\underline{3}$	2	2	$\underline{3}$	$\underline{3}$
$X_{2,3}$	0.80	2	6	2	2	2	4	$\underline{3}$
$X_{3,1}$	0.07	2	4	2	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{3,2}$	0.50	2	7	2	2	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{3,3}$	0.80	2	4	2	2	2	2	$\underline{3}$
$X_{4,1}$	0.07	2	4	2	2	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{4,2}$	0.50	2	9	2	2	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{4,3}$	0.80	2	10	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{5,1}$	0.07	2	4	2	2	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{5,2}$	0.50	2	$\underline{3}$	2	2	$\underline{3}$	$\underline{3}$	$\underline{3}$
$X_{5,3}$	0.80	2	$\underline{3}$	2	2	2	$\underline{3}$	$\underline{3}$

Table 3.3: The optimal number of clusters

2. We also note that the validity indices V_{ZLE} and V_{XIE} are very vulnerable to change in the OLR. They yield acceptable results when OLR is low. When OLR increases, they tend to favor smaller numbers of clusters. On the other hand, the results given by V_{ZLE} and V_{XIE} for $X_{4,1}$ and $X_{5,1}$ show that these indices fail even when OLR is low (0.07). We believe that elongated cluster shape and loose density have a stronger negative impact on the results yielded by V_{ZLE} and V_{XIE}.

3. The behavior of V_{FS} is special in that it reacts differently from all the other tested validity functions with respect to increasing OLR. It tends to favor a larger number of smaller clusters. Although it performs well with X_5, the experiments reported here seem to suggest that V_{FS} is also very sensitive to cluster shape and density.

4. The validity indices that yield consistently accurate and stable results are V_{WSJ}

and V_{RH}.

3.5 Discussion and Conclusion

The main contribution of this chapter is the introduction of the ridge curve concept, which allows us to establish a theory of overlap in mixtures of Gaussians. The significance of this theory is that it provides a mathematically rigorous way to explain the overlapping phenomenon and a computationally feasible way to calculate the degrees of overlap between the components of a mixture. It also provides a foundation for generating overlapped data sets for use in validating clustering and classification algorithms.

Comparing the overlap rate described in this thesis with conventional concepts of the distance of two Gaussian distributions such as Mahalanobis distance [83, 78], $((\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2))^{1/2}$, and Bhattacharyya distance [78], $\frac{1}{8}(\mu_1 - \mu_2)^T (\frac{\Sigma_1 + \Sigma_2}{2})^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \frac{|\Sigma_1 + \Sigma_1|}{2\sqrt{|\Sigma_1||\Sigma_2|}}$, we can make the following comments. All of these notions are related to the separability of two components. However, there are two major differences between the OLR and, say, the Bhattacharyya distance. One is that the OLR takes the a priori probability of each component into account instead of assuming that the two components have the same a priori probability. Although further studies are necessary, we believe that this property could potentially make the OLR a better measure of separability in various situations. Another more important difference is that not only is the OLR computed, but also the point in the space at which the OLR is reached is located. Apart from providing a "visual" interpretation and a better understanding of separability, this property is a key element for enabling the inverse operation, i.e., generating truthed data with a prescribed degree of overlap.

Chapter 4

Applications

In this chapter, we introduce two applications of the algorithms, developed in previous chapters. First, we propose a new method for feature selection in classification based on the new model selection clustering algorithm: **FBSA**. Second, we make our theory of cluster overlap to develop a hierarchical clustering algorithm for color image segmentation.

4.1 A Fuzzy Clustering Based Algorithm for Subset Feature Selection

The problem of subset feature selection for classification is defined as follows: Given a set of features, select the subset that performs the best under some classification system. Feature selection not only can reduce the cost of recognition by reducing the number of features that need to be collected, but in many cases can also provide better classification accuracy due to the finite sample size effect [9]. Using a subset of features can increase the understandability of the acquired knowledge. Feature selection can help data visualization by reducing the number of dimensions.

4.1.1 Introduction

Many methods are used for feature selection. Dash and Liu summarized these methods [84]. Feature selection involve: generating the subset of features and evaluating them. Three major strategies can be adopted in generating the subset of features: 1. Complete strategy involves examining all possible combinations of features, which becomes too expensive if the feature set is large; 2. Heuristic strategy uses certain guideline to control the selection processing; it is simple to implement and produces rapid results [85, 86]; 3. Random strategy selects features randomly (probability approach). Five types of function are often used to evaluate features subsets: 1. distance measures; 2. information measures; 3. dependence measures; 4. consistency measures; and 5. classification error rate measure.

Considering all of these methods and evaluation functions, the goal of feature selection can also be stated as finding the subset of features which is the most "structurally similar" to the original feature set. The "structural similarity" of two feature sets can be described by the cluster structure of two data sets. Dy and Brodley examined feature selection wrapped around expectation maximization (EM) clustering with order identification [87]. They introduced the clustering algorithm (EM) into the feature selection problem for unsupervised learning. For the classification problem, however, little attention has been paid to the role of clustering methods in feature selection. The difficulty stems from the complexity and inaccuracy of clustering algorithms when the number of clusters is not known.

Here, we propose a wrapper approach to feature selection using the efficient clustering technology, **FBSA**. The approach is based on the fact that the selected feature subset is "structurally similar" to the original feature set. Based on **FBSA**, we propose a novel algorithm for focusing on the structural similarity in feature selection process. We define a classification error rate for evaluating the subset of features. Extensive test results derived by applying the new algorithm to two artificial data sets and a real-world data sets are reported.

4.1.2 Feature Selection Procedure

We adopted the heuristic strategy for generating a feature subset. The goal of the procedure is to wrap the feature subset based on the clustering algorithm. Unlike the filter approach, which attempts to assess the merit of features from the data alone, the wrapper approach conducts a search for a good subset using an induction algorithm as part of the evaluation function [88]. The main idea of our algorithm is to evaluate each subset by a criterion, which defined by the classification error rate. The *Greedy* technique will be used in the search procedure. As we know, searching the entire feature subset space will lead to an $O(2^d)$ (d is the the number of entire features) computational problem. In order to by-pass the combinatorial explosion problem, we use a multi-step search process [84]. Each step tests remaining features and chooses the best one to add to the selected subset. The newly selected feature is the most "combinable" with those already selected. In other words, combining the new feature with the existing selected subset should lead to a lower classification error rate and this error should be the lowest among all the errors resulting from combining one non selected feature with the selected subset. The search process stops when adding any of the remaining feature to the selected subset would yield an increase in the classification error rate.

FSBC (Feature Selection Based Clustering) Algorithm:

```
1. Set selected subset $SS$ to empty, and $cr = 1$.
2. For any feature $f_i$, which is not in $SS$
    2.1 Let $T_i = SS \cup \{f_i\}$, and create a new subspace data set $SP_i$ using the features of $T_i$.
    2.2 Call FBSA on $SP_i$ to determine the number of clusters and produce clustering results.
    2.3 Compute the classification error rate $R(i)$ based on the subspace $SP_i$ (defined by $T_i$).
3. $R(j) = \min_{f_i \notin SS}\{R(i)\}$ with the lowest classification error rate.
4. if $R(j) < cr$ then $cr = R(j)$; $SS = T_j$; goto step 2; else output $SS$, stop.
```

The classification error rate in the feature selection algorithm is defined as follows: Suppose C is the number of classes in the original data set, S is a subset of features, $SP(S)$ is the subspace data set formed by S, $K(S)$ is the number of clusters in $SP(S)$, and $P(S, k, i)$ is the number of objects in cluster k of $SP(S)$ belonging to the class labelled i in the original data set. According to the majority rule,

$$CP(S, k) = \{j | P(S, k, j) = \max_{1 \leq i \leq C}\{P(S, k, i)\} \tag{4.1}$$

indicates that cluster k is related to class j or the main class label of cluster k is class j. Consequently

$$EC(S, k) = \sum_{i=1, i \neq CP(S,k)}^{n} P(S, k, i) \tag{4.2}$$

indicates the discrepancy between cluster k and its main class label. The classification error rate of S is defined as

$$R(S) = \sum_{k=1}^{K(S)} EC(S,k)/N \tag{4.3}$$

where N is the number of objects in the data set.

The value of $R(S)$ shows the accuracy of the representation of the original class information using the data set corresponding to the subset of features. The lower the value of $R(S)$, the better the representation is. This means that the cluster structure of the subspace data set is "similar" to the class structure of the original data set.

This structural similarity can be interpreted in a more intuitive way. Since the goal of feature selection is to better represent class information, we expect that the selected subset of features leads to a cluster structure with each cluster corresponding as closely as possible to a single class in the original data set. More than one cluster may correspond to a single class. However, the case where one cluster corresponds to multiple classes should be avoided. The classification error rate defined above allows us to grade each subset and distinguish the different cases. Obviously, this evaluation relies on accurate assessment of the structure of the data set corresponding to the subset of features. The use of an efficient clustering algorithm distinguishes our feature selection algorithm from existing ones.

4.1.3 Experimental Results

In this section, we will report experimental results on three data sets, of which one comes from the public domain, one is generated using a mixture of Gaussian distributions,

	subset(SS)	(f_i)	$R(s)$
step1		B_1	5/32
step2	$\{B_1\}$	B_0	5/32
step3	$\{B_1, B_0\}$	A_1	3/32
step4	$\{B_1, B_0, A_1\}$	C	5/32

Table 4.1: Selection results for DataSet1

and one is a real world data set. The first data set is CorrAl[89]. This data set has 32 instances. It contains two classes and six boolean features $(A_0, A_1, B_0, B_1, I, C)$, of which feature I is irrelevant, feature C is correlated to the class label 75% of the time, and the other four features are relevant to the boolean target concept: $(A_0 \wedge A_1) \vee (B_0 \wedge B_1)$. In [84], Dash and Liu tested the data using eight different feature selection algorithms. A few of them correctly select the actual subset (A_0, A_1, B_0, B_1), while most produce a subset including C or I. Although the data has not clear class structure, our algorithm results in a final selected feature subset including $\{B_1, B_0, A_1\}$. This result shows that all of the features selected are important, although one important feature is bypassed. Table 4.1 shows results at each selection step.

The second data set is generated using a mixture of Gaussian distributions. It contains 250 data points and has ten features $\{x_1, x_2, ..., x_{10}\}$. The first three features are significant. The subspace data set corresponding to the first three features $\{x_1, x_2, x_3\}$ is a mixture of five Gaussian components. The other features are as follows. $x_6 = 2 \times x_1, x_7 = 4 \times x_2, x_8 = 5 \times x_3$ are three relevant features. x_4 and x_5 are white-noise uniformly distributed variables. x_9 and x_{10} are "Gaussian noise". They are normal distributions and independent from each other. The class label is based on the first three features. There are 50 data points in each class. Due to the noise and excrescence features, classifying the data set using all features would result in a classification error

	subset(SS)	(f_i)	$R(s)$
step1		x_2	138/250
step2	$\{x_2\}$	x_3	111/250
step3	$\{x_2, x_3\}$	x_1	17/250
step4	$\{x_2, x_3, x_1\}$	x_7	90/250

Table 4.2: Selection results for DataSet2

rate of 178/250. Furthermore, this result does not indicate the class property of the data. By applying our new feature selection algorithm to the data set, the classification error rate is decreased remarkably, reaching 17/250. Table 4.2 shows the selection results. The selected feature subset is $\{x_2, x_3, x_1\}$.

The third experiment was done on feature sets extracted from an MSTAR small vehicle target/shadow image database. These features include moment, surface, shape, perimeter, Fourier descriptor, complexity, etc. We calculated a total of 20 features for each target and 20 features for the shadow. The feature vectors were previously grouped according to the orientations of the target. (Details about the image segmentation algorithm was presented in [58].) There are 3 classes of targets. The aim of the feature selection algorithm is to find appropriate features to aid in solving the target classification problem. Here we test 11 data sets, each of which contains the observation data from an orientation.

In this problem, we do not know which features are the best for targets. The features may play different roles in different target/ orientation combinations, so different target/ orientation combinations may need different features. Using all features in classification lead to inaccurate classification (average classification error rate is 44.6%) and high time cost.

Table 4.3 compares the time cost (seconds) and classification error using selected

	All Feature		Selected Feature		
	time cost	error rate (%)	selected feature	time cost	error rate (%)
data1	21	37	$\{f_{16}, f_4, f_1\}$	1.5	15
data2	144	39	$\{f_{16}, f_{18}, f_{40}, f_{26}\}$	14	25
data3	103	47	$\{f_{36}, f_{18}, f_{16}, f_{27}, f_{39}\}$	15	24
data4	36	47	$\{f_{16}, f_{36}, f_{19}, f_{18}\}$	3.6	18
data5	7	48	$\{f_{16}, f_1, f_{20}, f_{30}, f_2\}$	0.9	25
data6	4	34	$\{f_{19}, f_{15}, f_{37}\}$	0.3	21
data7	6	41	$\{f_{16}, f_{25}\}$	0.3	24
data8	16	55	$\{f_{16}, f_{25}, f_1\}$	1.3	34
data9	22	40	$\{f_{20}, f_{17}, f_{39}, f_{23}, f_5\}$	5	19
data10	53	56	$\{f_{16}, f_1, f_{38}, f_{40}, f_{24}\}$	5.2	29
data11	34	47	$\{f_{19}, f_5, f_1, f_{25}\}$	3.5	22
average	-	446	-	-	23

Table 4.3: Comparison of results for third experiment

feature subsets and using all features. Here, we use the nearest neighbor classifier as the basic classification algorithm and the same computer as mentioned in chapter 2. Using our new feature selection algorithm results in efficient feature subsets for classification in all data sets. The number of selected features is between 3 and 6 of the 40 features. This leads to reduced cost in terms of time and memory (85% lower using the selected features than when all features are used). The classification accuracy rises observably. The average classification error rate is down to 23% from 44.6%. Furthermore, the results shows that some features are important in practice. For example, f_{16} is selected in most data sets, f_1 and f_{18} are also often selected. This means they are relevant for the targets.

4.1.4 Concluding Remarks on the Feature Selection Algorithm

We have presented a wrapper approach to feature selection using fuzzy clustering, and proposed a new feature selection algorithm(FSBC) based on a clustering method.

The particularity of this algorithm can be summarized as follows: 1. the true number of clusters in the subspace data set, for use in determining the cluster structure of the subset of features; 2. the classification error rate when the subspace data set and the original data set contain different numbers of clusters (classes), for use in comparing the cluster structure information of a subspace data set and the class structure information of the original data set. We are currently carrying out an evaluation of the new algorithm, including comparison with existing algorithms and testing on different types of data sets.

4.2 Measuring Overlap Rate for Cluster Merging in a Hierarchical Approach to Color Image Segmentation

In color image segmentation, it is appropriate to study the cluster overlapping phenomenon with the hypothesis that cluster distributions can be described by mixtures of Gaussians. In this section, we present a new theory regarding cluster overlap, which allows for the computation of a cluster overlap rate that turns out to be a very good measure of similarity between clusters. Using this measure, we develop a new hierarchical algorithm for image segmentation that partially solves the problem of determining the best number of clusters. Experimental results demonstrate the effectiveness of the new algorithm.

4.2.1 Introduction

Image segmentation is an important first step towards image understanding. Clustering-based segmentation methods involve separating pixel features into clusters that correspond to homogeneous regions. Algorithms such as the k-means, the Fuzzy C-Means [27], EM [23] and many hierarchical algorithms [38] based on cluster merging are often used in image segmentation [90, 91, 92]. These algorithms are popular because they can be used to build robust and autonomous segmentation systems for dealing with large numbers of images in applications such as content-based image retrieval.

Many of these clustering algorithms assume implicitly that the data distribution model is a mixture of Gaussians. This offers a good trade-off between the model's complexity and its capacity to approximate the probability density function (pdf) of the image pixels (feature vectors). In image segmentation, the smoothing operation (preprocessing) often applied before the clustering operation makes the mixture of Gaussians hypothesis even more realistic. There is a close relationship between the components in the mixture model and the clusters in the feature space. Cluster analysis contributes to image segmentation by identifying clusters in the input data in order to accurately estimate the parameters of the mixture including the number of components, the parameters of each component (mean and covariance matrix) and the mixing parameters (prior probability of each component).

Hierarchical clustering techniques are also frequently used in image segmentation. They generate a nested series of partitions by merging clusters (agglomerative approach) or splitting them (divisive approach), based on a measure of similarity. Measuring the similarity between clusters is the key to hierarchical algorithms. Typical similarity measures include minimum distance (single-link), maximum distance (complete-link) and average distance (average-link) [37]. All of these distances are based on direct comparison between points in two clusters and do not take into account the distribution model of each cluster. The time and space complexity of a hierarchical algorithm is typically higher than for a partition-based algorithm. However, hierarchical algorithms are more versatile than partitioning algorithms. They often work well on data sets with non-isotropic clusters. Like many partition-based algorithms, hierarchical algorithms also require that the number of clusters be given in advance.

The problems of dealing with overlapping clusters and estimating the number of clusters share a common root. Both require a mathematically sound criterion for deciding whether overlapping clusters are distinguishable from each other. Current understanding of the overlapping phenomenon in the multi-dimensional case is very intuitive. There is no simple measure that characterizes the degree of overlap between components as a function of the mixture parameters. This is why work on the design and evaluation of clustering algorithms is often carried out on an empirical basis. The similarity measures used in hierarchical clustering, as mentioned in the previous paragraph, are an example. Evaluation of clustering algorithms is another example. Few authors evaluate their algorithms in a formal way, according to the basic mixture of Gaussians hypothesis regarding data distribution, which is fundamental to the design of their clustering algorithms.

Here, we propose a hierarchical approach for color image segmentation. The approach is based on the measurement of the similarity between two clusters. The theory of cluster overlapping, which we presented in previous chapter, provides a natural way to merge two clusters. The new algorithm starts from a initial number of clusters and

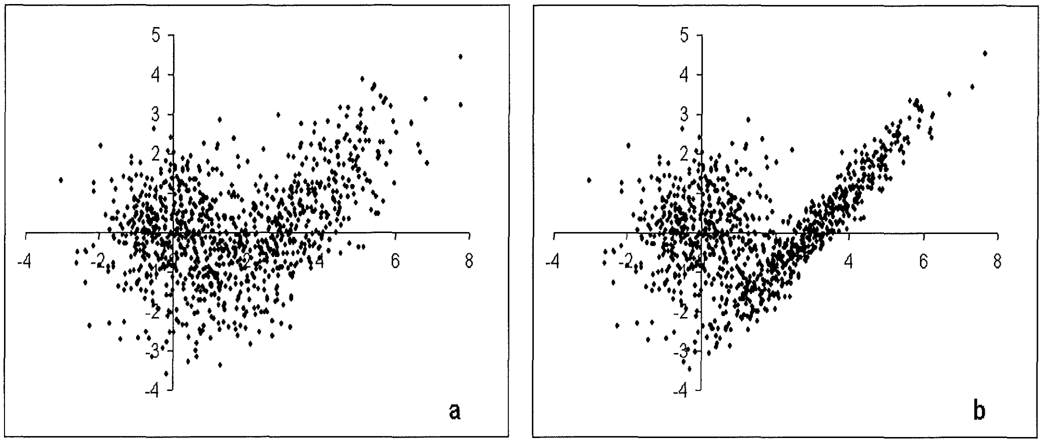


Figure 4.1: Generated clusters for comparing average-distance-based and OLR-based similarity measures

merges the clusters with higher overlap rate for decreasing the number of clusters to its true value.

4.2.2 Overlap Rate and Distance between Two Clusters

The overlap rate proposed in the previous section can be used directly as the similarity measure in a hierarchical clustering algorithm. In fact, the conventional similarity measures based on the minimum distance or the maximum distance are clearly not appropriate under the hypothesis that each cluster follows a Gaussian distribution (has an ellipsoidal shape). The following example shows why OLR is also a better similarity measure than the average distance.

Figures 4.1-a and 4.1-b both contain artificially generated clusters. The cluster centered at the origin is the same in both figures. The other cluster in each of the two figures has the same center, orientation and length on the principal axis. They differ only in the width on the secondary axis. The overlap rates between the two clusters in each figure are very different (OLR=0.811 in Figure 4.1-a vs. OLR=0.388 in Figure 4.1-b), while the average distance is very similar in the two figures (3.578 vs. 3.541). Thus, according to the average distance, we obtain the strange conclusion that the two clusters in Figure 4.1-b are more similar than the clusters in Figure 4.1-a.

Another advantage of using the overlap rate as a similarity measure, compared to distance measures, is that the value of OLR is normalized between 0 and 1. This makes the values of similarity between clusters more comparable. More importantly, by examining the maximal OLR, we can obtain an idea of the minimum separation between clusters at each phase of iteration of a clustering procedure. This could be used, in turn, as a condition to stop the clustering process and to determine the number of clusters.

There is, however, a problem with using the overlap rate as the similarity measure. Since it requires estimation of the mean and covariance matrix, each cluster must have at least a minimum number of data points. Therefore, a hierarchical algorithm using OLR as a similarity measure cannot start with one data point for one cluster, as is the case in a conventional approach. A solution to this problem could be to start with a conventional approach using the average distance as the similarity measure, and then switch from the distance measure to OLR when the number of data points in each cluster is sufficient. We have chosen another approach to this problem. It is known that running the k-means algorithm, using a large number of clusters and a random procedure to initialize the cluster centers, can yield a set of clusters that provides good coverage of the data set. The hierarchical clustering techniques can then be applied to the results of the K-Means algorithm.

4.2.3 OLR-based Merging Strategy in Hierarchical Clustering Algorithm

In designing of the new algorithm, the issues of determining the number of clusters and of merging efficiency received most of our attention. In general, we could control the number of clusters by setting a minimum threshold value for OLR and running an iterative procedure for cluster merging, each time choosing the cluster pair having the maximum OLR value exceeding the threshold. In image segmentation, one usually wants to set the maximal number of clusters (number of regions with different characteristics), C_{max}, and hope that the algorithm can find the best number between 1 and C_{max}. It is difficult for a user to set the threshold so that the merging process runs until the number of remaining clusters falls below C_{max}. On the other hand, nor is it a good idea to use the maximal number alone as the controlling parameter for merging process, since the merging would always stop at $C_{max} - 1$ clusters. The efficiency problem is related to the number of pairs of clusters that are merged in a single iteration phase.

Algorithm COLRM (Clustering using OLR-based merging)

1. Input a data set

2. Enter values for the parameter C_{init}, C_{max} and δ (a parameter to control the OLR level).

3. Set $C = k = C_{init}$ and cluster the data set by the K-Means algorithm (C is used to keep track of the number of clusters at each phase).

4. Calculate the overlap rate: O_{ij}, between each pair of clusters ($1 <= i < j <= C$)

using **Algorithm COLR** and calculate the maximum overlap rate

$$CurrMaxOLR = \max_{1 \leq i \leq j \leq C} \{O_{ij}\} \tag{4.4}$$

5. Select and merge all the possible candidates of cluster pairs:

   (a) A pair of clusters $(i,\ j)$ is a candidate if its O_{ij} satisfies:

   $$CurrMaxOLR - \delta < O_{ij} \leq CurrMaxOLR \tag{4.5}$$

   (b) A candidate is selected for merging is if its O_{ij} satisfies:

   $$O_{ij} = \max_{k \geq j, l \leq i} \{O_{ik}, O_{lj}\} \tag{4.6}$$

   (c) Merging two clusters is performed by grouping all the data points of the clusters in one subset and recalculating the mean, covariance matrix and the mixing parameter according to Eq.3.19.

6. Update the current number of clusters C.

7. Decide whether the current OLR threshold should be further reduced:

   (a) If $(C > C_{max})$, then goto Step4.

   (b) - Otherwise, goto Step8.

8. Calculate the overlap rate O_{ij} between each pair of clusters $(1 <= i < j <= C)$, and then goto Step5 if the expression Eq.4.5 holds at least for one pair of clusters, otherwise, output the clusters.

The algorithm contains three input parameters that can be easily set by the user.

- C_{init}: the initial number of clusters for the run of k-means algorithm. Any number smaller than n/d could possible be used for C_{init}. In image segmentation, n/d can be a very large number. Using a number that is too large could significantly increase the runtime of the k-means algorithm. In our experience, any value between 30 and 100 is a good choice for images whose size ranges from 128×128 to 512×512.

- C_{max}: the maximum number of clusters in the final segmented image. Usually, the condition, $C_{max} < C_{init}$, holds.

- δ: a parameter for controlling the merging process. The merging procedure selects all pairs whose OLR value falls within the interval $(CurrMaxOLR - \delta, CurrMaxOLR)$ as candidates for merging. δ is a more sensitive parameter than the two previous ones. It has some impact on the time efficiency and on the number of final clusters obtained. According to our experience, any value between 0.05 and 0.15 is a good choice. In this paper, we have used $\delta = 0.1$ in all tests of the proposed algorithm.

4.2.4 Application to Color Image Segmentation

Next, we report several preliminary results of applying the proposed algorithm to color image segmentation. Both synthetic and real images were used. In order to show the effectiveness of the algorithm, the pre-processing of each image was limited to minimum. In fact, no pre-processing was applied to synthetic images. The transformation to the CIE (L, u, v) color coordinate system is applied to real images in order to unify the color space. Only color features, *i.e.* (R, G, B) from synthetic images and (L, u, v) from real

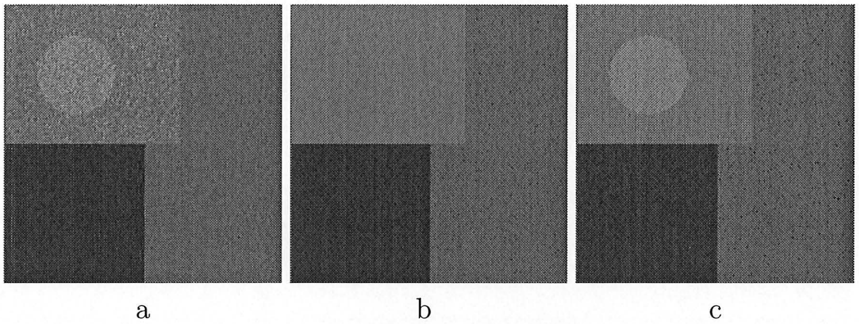


Figure 4.2: Comparison between the k-means algorithm and the new algorithm. (a) is the original image; (b) is the image obtained by applying the k-means algorithm with k=4 (which is the known number of classes in this case); (c) is the image obtained by the new algorithm (the number of clusters, which is 4, is obtained by the algorithm).

images are used as input to the clustering algorithm. Throughout the experiments the three input parameters are set as follows: $C_{init} = 30; C_{max} = 6$ and $\delta = 0.1$.

4.2.4.1 Experiments with Synthetic Images

The Fig.4.2 and Fig.4.3 show experiments with two synthetic images. Fig.4.2-a shows the original image generated using a mixture of four Gaussian components. The main difficulty with this image is with the "circle" and its "background" in the upper left part of the image. The Gaussian components corresponding to these two "objects" are quite close to each other. The shape of each component is elongated. They are parallel along the principal axis. However, the two components are well separated from each other (OLR is very small). Partition-based algorithms such as the k-means cannot distinguish between the two clusters because of their proximity if the number of clusters is initialized to be the true number of components. Using the proposed algorithm, the two objects are well separated because the two components are well "covered" by small

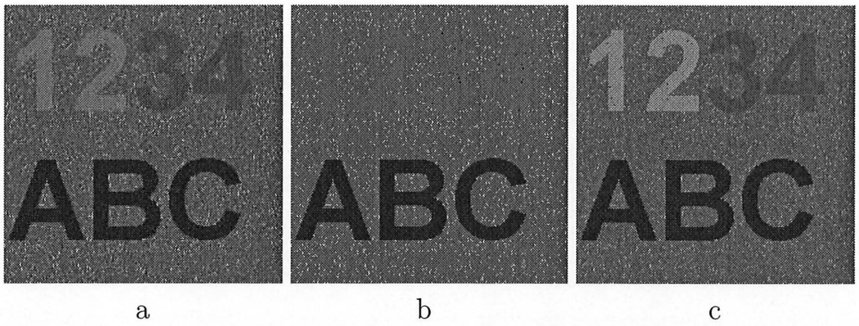


Figure 4.3: Comparison between the k-means algorithm and the new algorithm. (a) is the original image; (b) is the image obtained by applying the k-means algorithm with k=4 (which is the known number of classes in this case); (c) is the image obtained by the new algorithm (the number of clusters, which is 6, is obtained by the algorithm).

Component Center	Covariance Matrix of the Component		
100	1	0	0
150	0	400	0
150	0	0	25

Table 4.4: Component parameters of the "circle" object in Fig.4.2-a.

clusters after the first step of the K-Means, and then, the merging process takes place only between clusters covering the same object. Fig.4.2-b and -c, respectively, show the segmentation results using only the k-means algorithm (with k=4) and using the new algorithm. Detailed information regarding the two objects in the three images is given in Tables 4.4-4.8.

Finally, Fig.4.3 depicts another difficulty in image segmentation that is overcome by

Component Center	Covariance Matrix of the Component		
40	1	0	0
150	0	400	0
150	0	0	25

Table 4.5: Component parameters of the upper-left "background" object in Fig.4.2-a.

Component Center	Covariance Matrix of the Component		
51.50	577.11	1.42	0.41
149.29	1.42	398.82	-0.24
149.51	0.41	-0.24	9.93

Table 4.6: Cluster parameters of the upper-left object in Fig.4.2-b, obtained by the K-means algorithm .

Component Center	Covariance Matrix of the Component		
99.50	1.04	0.56	0.17
149.40	0.56	394.35	-1.49
149.55	0.17	-1.49	24.54

Table 4.7: Cluster parameters of the "circle" object in Fig.4.2-c, obtained by the proposed algorithm .

Component Center	Covariance Matrix of the Component		
39.50	1.06	-0.30	0.003
149.26	-0.30	400.22	0
149.50	0.003	0	6.29

Table 4.8: Cluster parameters of the upper-left object in Fig.4.2-c, obtained by the proposed algorithm .

the new algorithm. The component "34" lies between component "12" and component "ABC". All three are small components compared to the "background" component. In fact, some noise in the "background" object forms clusters that are as big as components "34" and "12". Depending on the initialization, the k-means algorithm (with k=4) either groups "34" with "12" (case shown in Fig.4.3-b) or with "ABC". The proposed algorithm has no problem in distinguishing between these clusters although it does create two more clusters from the "background" objects.

4.2.4.2 Experiments with Real Images

For the Lena image, the k-means algorithm is initialized using the number of clusters obtained by the new algorithm. The results are shown in Fig.4.4. By examining the results in Fig.4.4-b and -c, we clearly see that the new algorithm performs well in merging gradually changing pixels within each region and in preserving the natural boundaries between regions. These properties are also reflected in the results with the three images shown in Fig.4.5.

4.2.5 Results and Discussion

In this section, we report a study of hierarchical clustering based on a new concept of the overlap rate between clusters. Computation of OLR is facilitated by using the theory of cluster overlapping we developed recently. We have presented several arguments on why the overlap rate may be a better similarity measure for comparing clusters than distance-based measures. The overlap rate allows us to design a novel hierarchical algorithm that is capable of automatically determining the number of clusters. The

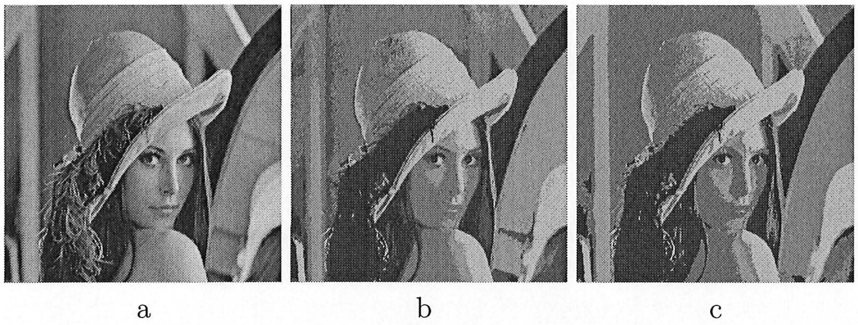


Figure 4.4: Comparison between the k-means algorithm and the new algorithm. (a) is the original image; (b) is the image obtained by applying the k-means algorithm with k=5 (which is the known number of classes in this case); (c) is the image obtained by the new algorithm (the number of clusters, which is 5, is obtained by the algorithm).

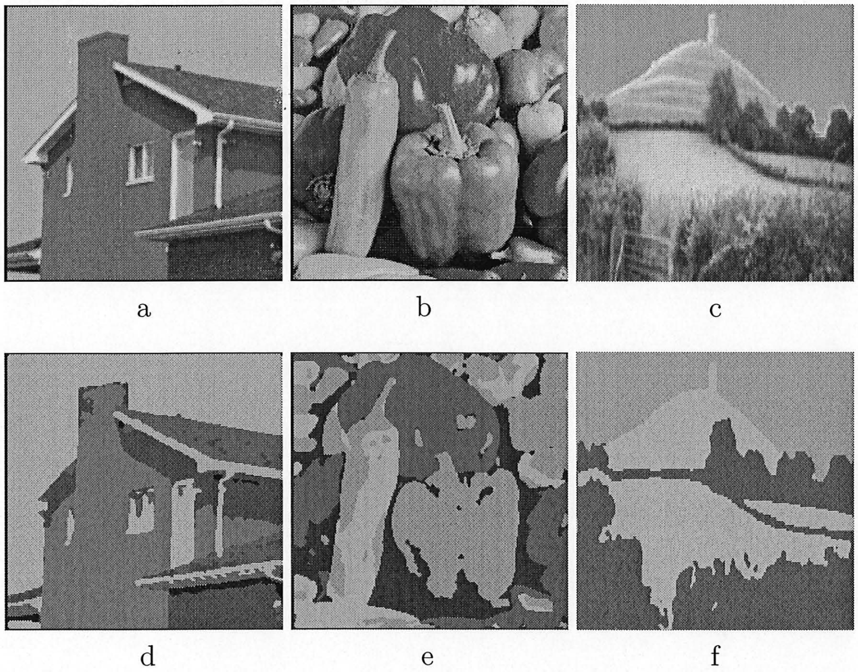


Figure 4.5: Examples showing the performance of the proposed algorithm. (a), (b) and (c) are original images and (d) (6 clusters), (e) (4 clusters), and (f) (3 clusters) are the corresponding segmented images yielded by the proposed algorithm.

preliminary experiments demonstrate the effectiveness of the new algorithm. Several additional comments are worth mentioning:

- One might want to compare the overlap rate with the Bhattacharyya distance, which is also related to the distance between (or separability of) two components. There are two major differences between the overlap rate and the Bhattacharyya distance. The overlap rate takes into account the prior probability of each component instead of assuming they are equal for the two components. Moreover, it is a geometrical concept. The point where the maximal overlap rate is reached is located. The information can be further explored for better segmentation of the data set.

- The new algorithm is quite time-efficient. In fact, the time required to run this algorithm is equivalent to running the k-means algorithm in the first step. The time for the following steps related to cluster merging is negligible.

- The new algorithm can be adapted to deal with cluster models other than mixtures of Gaussians. For instance, chain-like or ring-like clusters can be obtained by allowing representation of each cluster by multiple centers and covariance matrices and by using more restrictive conditions for cluster merging.

- As an algorithm for image segmentation, it can be used with different features including color and texture jointly or separately in order to obtain better segmentation results.

Conclusion

In this thesis, we have proposed efficient model selection algorithm for determining the number of clusters. We established a theory on overlapping phenomenon between clusters. The new model selection algorithm improves both the accuracy of the number of clusters and time cost. The theory of overlapping allows to characterize the phenomenon of overlapping between clusters. Both have important impact in real applications. We summarize our achievements before discussing future research directions.

1. Achievements

1.1 Determining the Number of Clusters

We have developed a new validity index for measuring the results of a clustering and a new model selection algorithm for determining the number of clusters for a given data set. The algorithm has been used to develop a feature selection algorithm and an image segmentation system [1]. Our major contributions are as follows:

[1] The clustering algorithm of in Chapter 2. has been used straight forwarding in developing an image segmentation system. Details have not been mentioned in this thesis.

- **Yielding accurate numbers of clusters.** There are a lot of validity indices designed for determining the number of cluster. However, for the data sets with overlapped clusters, they often fail to yield the true number of clusters. In Chapter 2, we have presented a new cluster validity, which is a function of the original data, cluster centers and membership. A paper titled "**A New Validation Index for Determining the Number of Clusters in a Data Set**", describing this work in addressing the number of clusters problem, was published in Proceeding of IJCNN'2001 [54].

- **Efficient model selection algorithm.** Determining the number of cluster is often based on "trial and error" approach. The "trial and error" approach is affected randomly by initialization in the "trial" procedure. We investigated into the effect of random initialization. A paper titled "**New FCM-based Algorithms for Finding the Number of Clusters**", describing our achievements in addressing the time efficiency and stability of model selection algorithm, was published in Proceeding of Proceedings of ICONIP2001 [93].

A full paper, titled "**FCM-based Model Selection Algorithm for Determining the Number of Clusters**", integrating the new validity index and the model selection algorithm, was published in international journal "Pattern Recognition" [6].

1.2 Feature Selection Based on Fuzzy Clustering

We deals with a wrapper approach to the problem of feature selection for classification. Based on fuzzy clustering, we develop a new algorithm that operates by testing the error between the cluster structure of the subspace data set and the class structure of the original data set. A paper titled "**A Fuzzy Clustering Based Algorithm for Feature Selection** has been published in IEEE: The First International Conference on Machine Learning and Cybernetics (ICMLC), Nov., 2002 [94].

1.3 Investigating the Phenomenon of Overlap between Clusters

We have established a theory on the phenomenon of overlap between components (clusters) in a mixture model. The novel concept of ridge curve and its properties is the foundation of the new theory. A short paper, titled "**Distinguishing Between Overlapping Components in Mixture Models**", has been published in The 2nd IASTED International Conference on NEURAL NETWORKS AND COMPUTATIONAL INTELLIGENCE, Feb., 2004, Switzerland [44]. A research report, titled "**A Theory on Distinguishing Overlapping Components in Mixture Models**", containing a detailed description of the theory and its application in generating truthed data sets, was made [77]. This report was recently extended and a complete version was submitted to IEEE trans. Systems, Man and Cybernetics.

1.4 Hierarchical Approach to Color Image Segmentation Based on the Overlap Theory

Based on the theory on the overlapping phenomenon, we have designed a novel hierarchical approach to color image segmentation. Merging the similar "objects" (clusters or components) is the main idea of the approach. The overlap rate between two "objects" is the main evidence for merging them. A paper, titled "**Measuring Overlap-Rate for Cluster Merging in a Hierarchical Approach To Color Image Segmentation**" has been accepted by **International Journal Fuzzy System** [80].

1.5 Validation of Validity Index

Based on a series of generated truthed data sets, which contains certain overlapped clusters, we present a new approach for the objective evaluation of validity indices and clustering algorithms. A paper, titled "**An Objective Approach to Cluster Validation**" has been submitted to **Pattern Recognition Letters** [95].

2. Future Research

Future work will focus on the applicability of our new validity index and new theory on cluster overlapping and the extension of the new theory to other distribution models.

- **Sensitivity to cluster overlapping of new validity.** An extensive evaluation of the new index and existing indices in terms of sensitivity to cluster overlapping is needed for data-mining applications. Such an evaluation can only be carried out with generated data sets, since they allow us to test clustering algorithms in a more controlled way. In Chapter 3, we have proposed a formal definition of the overlap rate between clusters and have developed methods for automatically generating data sets. It will be used to evaluate existing indices and algorithms (a simple result is reported in Section 3.4.3).

- **Application of the new theory to cluster data with non-spherical clusters.** The study on the cluster structure of a given data set is very important for designing efficient clustering algorithm. The cluster structure of a data set includes the shape of clusters and the relationship between clusters. In a general clustering algorithm, a spherical cluster is represented by one cluster center. For the complex clusters, such as line-cluster, ring-cluster and other non-spherical clusters, these algorithms often failed. An intuitive way for solving the problem is to combine multiple small sub-clusters and presenting the cluster by multi-centers. In [32], C. W. Tao proposes an algorithm for handling complex clusters. In the algorithm, multiple centers are used to represent the non-spherical shape of clusters. In [42], Karypis *et al.* design an algorithm that uses dynamic modelling in cluster aggregation. This is a hierarchical approach by merging small sub-clusters. These methods ignore a fact that some sub-clusters could come from a same distribution so that they should be merged to one. How to judge these sub-clusters is the key to understand the "cluster structure". We propose to investigate into the cluster structure. Based on the theory on cluster overlapping, the research on the parameters of the model will help determining the condition for merging multi sub-clusters to one. By further investigating the covariance matrix of a component in the Gaussian mixture, we can determine the orientation of the component. We will investigate into the conditions under which multiple clusters can be merged or combined in order to form cluster of complex shapes. We will also investigate how such an approach can be combined with projected cluster so that we can deal with high dimension data more efficiently.

Bibliography

[1] M. Menard and M. Eboueya, "Extreme physical information and objective function in fuzzy clustering," *Fuzzy Sets and Systems*, vol. 128, pp. 285–303, 2002.

[2] E. Anderberg, "The cluster analysis for applications," in *Academic Press. Inc.*, (New York, NY), 1973.

[3] V. Natalia, J. H. Brian, and L. S. Steven, "A Clustering Method for Repeat Analysis in DNA Sequences," *Genome Biology*, vol. 2, August 2001.

[4] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *4th ICAPRDT Proc.*, pp. 137–143, 1999.

[5] R. Agrawal, J. Gehrke, and D. Gunopulos, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proc. ACM-SIGMOD Int. Conf. Management of Data(SIGMOD'98)*, (Seattle, WA), pp. 94–105, 1998.

[6] H. Sun, S. Wang, and Q. Jiang, "FCM-based model selection algorithm for determining the number of clusters," *Pattern Recognition*, vol. 37, no. 10, 2004.

[7] Y. Chen, L. Qiu, W. Chen, L. Nguyen, and R. Katz, "Clustering web content for efficient replication," in *Proceedings of the 10 th IEEE International Conference on Network Protocols (ICNP02)*, 2002.

[8] J. Han and M. Kamber, *Datamining: Concepts and Techniques.* San Francisco: Morgan Kaufmann Publishers, 2001.

[9] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 152–157, 1997.

[10] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* NJ: Prentice-Hall, Englewood Cliffs, 1988.

[11] C. C. Aggarwal, "Finding generalized projected clusters in high dimensional spaces," in *Proceedings of ACM SIGMOD Conference, 2000*, 2000.

[12] E. Diday, "The dynamic cluster method in non-hierarchical clustering," *J. Comput. Inf. Sci.*, 1976.

[13] R. Michalski, R. Steep, and E. Diday, "Automated construction of classifications: conceptual clustering versus numerical taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5*, vol. 5, pp. 396–409, 1983.

[14] A. Hlaoui, H. Sun, and S. Wang, "A graph clustering algorithm with applications to content-based image retrieval," in *Proceedings of IEEE ICMLC 2002*, (Beijing, China), 2002.

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

[16] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, Canada, 1990.

[17] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," (San Francisco, CA), pp. 91–99, Proc. 15th International Conf. on Machine Learning, Morgan Kaufmann, 1998.

[18] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 283 – 304, September 1998.

[19] J. Lozano and P. L. J.M. Pena, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern Recognition Lett.*, vol. 20, pp. 1027–1040, 1999.

[20] A. Likas, N. Vlassis, and J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[21] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[22] J. Bentley, "Multidimensional binary search trees used for associative searching," *Comm. ACM*, vol. 18, pp. 509–517, 1975.

[23] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.

[24] S. L. Lauritzen, "The em algorithm for graphical association model with missing data," *Computational Statistics and Data Anlysis*, vol. 19, pp. 191–201, 1995.

[25] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.

[26] J. Bezdek, "Fuzzy mathematics in pattern classification." Ph.D. Dissertation, Cornell University, 1973.

[27] J. C. Bezdek, *Pattern Recogniation with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[28] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[29] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *Fuzzy Systems*, vol. 1, pp. 98–109, May 1993.

[30] F. Hoppner and F. Klawonn, "Improved fuzzy partitions for fuzzy regression models," *International Journal of Approximate Reasoning*, vol. 32, pp. 85–102, 2003.

[31] L. Romdhane, B. Ayeb, and S. Wang, "On computing the fuzzifier in —flvq: a data driven approach," *International Journal of Neural Systems,*, vol. 12, no. 2, pp. 149–157, 2002.

[32] C. Tao, "Unsupervised fuzzy clustering with multi-center clusters," *Fuzzy Sets and Systems*, vol. 128, pp. 305–322, 2002.

[33] A. Devilleza, P. Billaudelb, and G. Lecolierc, "A fuzzy hybrid hierarchical clustering method with a new criterion able to find the optimal partition," *Fuzzy Set and System*, vol. 128, pp. 323–338, 2002.

[34] N. Belacel, P. Hansen, and N. Mladenovic, "Fuzzy j-means: a new heuristic for fuzzy clustering," *Pattern Recognition*, vol. 35, pp. 2193–2200, 2002.

[35] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy.* London, UK: Freeman, 1973.

[36] G. Nagy, "State of the art in pattern recognization," in *Proc. IEEE 56*, pp. 836–862, 1968.

[37] B. King, "Step-wise clustering procedures," *J. Am. Stat. Assoc.*, vol. 69, pp. 86–101, 1967.

[38] R. A. Baeza-Yates, *Introduction to data structures and algorithms related to information retrieval*, pp. 13–27. Prentice-Hall, 1992.

[39] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *In KDD Workshop on Text Mining*, (Boston, MA, USA), 2000.

[40] T. Zhang, R. Ramakrishnan, and M.livny, "Birch: An efficient data clustering method for very large database," in *In Proc. 1996 ACM-SIGMOD Int. Conf. Managerment of Data*, (Montreal, Canada), pp. 103–114, 1996.

[41] S. Guha, R. Rastogi, and K. Shi, "CURE: an efficient clustering algorithm for large databases," in *Proc. of the ACM SIGMOD Int '1 Conf. Management of Data*, pp. 73–84, 1998.

[42] G. Karypis, E. H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[43] W. Day, *Complexity Theory: An Introduction for Practitioners of Classification.* World Science Co. Inc., River Edge NJ, 1992.

[44] H. Sun and S. Wang, "Distinguishing between overlapping components in mixture models," in *International Conference on Neural Networks and Computational Intelligence*, pp. 102–108, 2004.

[45] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining (KDD-96)*, pp. 232–237, 1997.

[46] M. Ankerst, M. Breunig, H. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *Int. Conf. Management of Data and Symposium on Principles of Database Systems*, (Philadelphia, Pennsylvania, USA), 1999.

[47] A. Hinneburg and D. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD-98)*, (NY, USA), pp. 58–65, 1998.

[48] W. Wang, J. Yang, and R. Muntz, "String: A statistical information grid approach to spacial data mining," in *Proc. Int. Conf. Very Large Database(VLDB97)*, (Athens Greece), pp. 186–195, 1997.

[49] J. C. Sheikholeslami, S. Chatterjee, and A. Zhang, "Wave:cluster: A multiresolution clustering approach for very large spacial database," in *Proc. Int. Conf. Very Large Database(VLDB98)*, (NY USA), pp. 428–439, 1998.

[50] C. Fraley, "Algorithm for model-based gaussian hierachical clustering," *SIAM J. Sci. Comput*, vol. 20, no. 1, pp. 270–281, 1998.

[51] N. Bouguila, D. Ziou, and J. Vaillancourt, "Maximum likelihood estimation of the generalized dirichlet mixture," *Accepted in IEEE Transactions on Image Processing*, 2004.

[52] A. Zhou and W. Q. et. al., eds., *A Hybrid Approach to Clustering in Very Large Databases*, (Hong Kong, China), 2001.

[53] J. C. Bezdek, *Chapter F6: Pattern Recognition in Handbook of Fuzzy Computation.* IOP Publishing Ltd, 1998.

[54] S. Wang, H. Sun, and Q. Jiang, "A new validation index for determining the number of clusters in a data set," in *Proceedings of IJCNN*, (Washington D.C., USA), pp. 1852–1857, July 2001.

[55] J. Costa and M. Netto, "Estimating the number of clusters in multivariate data by self-organizing maps," *International Journal of Neural Systems*, vol. 9, no. 3, pp. 195–202, 1999.

[56] B. Everitt, *Cluster Analysis, 3rd edition.* Edward Arnold, 1993.

[57] Y. Qu and Z. Feng, "A simulation study on determining the number of components in mixture distributions," in *Mixtures 2001:Recent Developments in Mixture Modelling*, 2001.

[58] E. Aitnouri, F. Dubeau, S. Wang, and D. Ziou, "Controlling mixture component overlap for clustering algorithms evaluation," *J. of Pattern Recog. andImage Analysis*, vol. 12, no. 4, pp. 331–346, 2002.

[59] A. Gordon, *Classification (2nd edition).* Chapman and Hall/CRC press, 1999.

[60] J. Pena, J. Lozano, and P. Larranaga, "An empirical comparison of four initialization methods for the k-means algorithm," *Pattern Recognition Letters*, vol. 20, pp. 1027–1040, 1999.

[61] Y. Fukuyama and M. Sugeno, "A New Method of Choosing the Number of Clusters for the Fuzzy C-means Method," in *Proceedings of 5th Fuzzy Systems Symposium*, pp. 247–250, 1989.

[62] X. Xie and G. Beni, "A Validity Measure for Fuzzy Clustering," IEEE *Transactions on Pattern Analysis and Machine Intelligence (*PAMI*)*, vol. 13, no. 8, pp. 841–847, 1991.

[63] H. Rhee and K. Oh, "A Validity Measure for Fuzzy Clustering and Its Use in Selecting Optimal Number of Clusters," in *Proc. of the IEEE International Conference on Fuzzy Systems(FUZZ-IEEE '96) New orleans*, pp. 1020–1025, 1996.

[64] N. Zahid, O. Abouelala, M. Limouri, and A. Essaid, "Unsupervised fuzy clustering," *Pattern Recognition Letters*, vol. 20, pp. 123–129, 1999.

[65] M. Rezae, B. Letlieveldt, and J. Reiber, "A new cluster validity index for the fuzzy c-means," *Pattern Recognition Letters*, vol. 19, pp. 237–246, 1998.

[66] T. Gonzalez, "Clustering to Minimize and Maximum Intercluster Distance," *Theoretical Computer Science*, vol. 38, pp. 293–306, 1985.

[67] N. Pal and J. Bezdek, "On Cluster Validity for the Fuzzy C-Means Model," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 3, pp. 370–390, 1995.

[68] S. Halgamuge and M. Glesner, "Neural networks in designing fuzzy systems for real world applications," *Fuzzy Sets and Systems*, vol. 65, no. 1, pp. 1–12, 1994.

[69] Y. Yeung, C. Fraley, A. Murua, A. Raftery, and W. Ruzzo, "Model based clustering and data transformation for gene expression data," *Bioinformatics*, vol. 17, pp. 977–987, 2001.

[70] K. Mosler, "Mixture models in econometric duration analysis," in *Mixtures 2001: Recent Developments in Mixture Modelling*, 2001.

[71] J. Puzicha, J. Buhmann, and T. Hofmann, "Discrete mixture models for unsupervised image segmentation," in *DAGM-Symposium*, 1998.

[72] J. Rajapakse, J. Giedd, and J. Rapoport, "Statistical approach to segmentation of single-channel cerebral mr images," 1997.

[73] S. Kirshner, I. Cadez, P. Smyth, and E. Pazand, "Probabilistic modelbased detection of bent-double radio galaxies," in *In Proc. 16th Int. Conf. on P.R 2*, pp. 499–502, 2002.

[74] M. Whindham and A. Cutler, "Information ratios for validating mixture analysis," *J. of the American Satistical Association*, vol. 87, pp. 1188–1192, 1992.

[75] C. Wallace and D. Boulton, "An information measure for classification," *Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968.

[76] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[77] H. Sun, "A theory on distinguishing overlapping components in mixture models," *Research Report, DMI, University of Sherbrooke, No 345*, Nov. 2003.

[78] K. Fukunaga, *Introduction to Statistical Pattern Recognition, 2nd edition.* Academic Press, Inc., 1990.

[79] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* 2001.

[80] S. Wang and H. Sun, "Measuring overlap-rate for cluster merging in a hierarchical approach to color image segmentation," *International Journal Fuzzy Systems*, vol. 6, no. 3, pp. 147–156, 2004.

[81] G. Milligan, "An examination of the effect of six types of error perturbation on fifteen clustering algorithms," *Psychometrica*, vol. 45, no. 3, pp. 325–342, 1980.

[82] B. Noble, *Applied Linear Algebra.* Prentice-Hall, Inc, Englewood Cliffs, N.Y., 1977.

[83] G. McLanchlan and K. Basford, *Mixture Models.* Marcel Dekker, Inc. N.J., 1987.

[84] M.Dash and H.Liu, "Feature Selection for Classification ," *Intelligent Data Analysis*, vol. 1, pp. 131–156, 1997.

[85] K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of Ninth National Conference on Artificial Intelligence*, 1992.

[86] C. Cardie, "Using decision trees to improve case-based learning," in *In Proceedings of Tenth International Conference on Machine Learning*, pp. 25–32, 1993.

[87] J. Dy and C. Brodley, "Feature subset selection and order identification for unsupervised learning," in *Proceedings of the 17th International Conference on Machine Learning*, pp. 247–254, 2000.

[88] R. Kohavi and G. John, "The wrapper approach," in *Feature Selection for Knowledge Discovery and Data Mining, Kluwer International Series in Engineering and Computer Science, Chap. 1*, 1998.

[89] G. John, R. Kohavi, and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 121–129, 1994.

[90] K. Krishna and M. N. Murty, "Genetic k-means algorithm," *IEEE Trans. SMC - Part B*, vol. 29, pp. 433–439, June 1999.

[91] Y. Lim and S. Lee, "On the color image segmentation algorithm based on the thresholding and fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 1235–1252, 1990.

[92] T. Uchiyama and M. Arbib, "Color image segmentation using competitive learning," vol. 16, no. 12, 1994.

[93] H. Sun, S. Wang, and Q. Jiang, "New fcm-based algorithms for finding the number of clusters," in *Proceedings of ICONIP2001*, (Shanghai, CHINA), pp. 564–569, Nov. 2001.

[94] H. Sun, S. Wang, and M. Mei, "A fuzzy clustering based algorithm for feature selection," in *Proceedings of ICMLC2002*, (Beijing, CHINA), Nov. 2002.

[95] M. Bouguessa, S. Wang, and H. Sun, “An objective approach to cluster validation,” *submitted to Pattern Recognition Letters*, 2004.