MIXTURE MODEL CLUSTERING
FOR VERY LARGE DATA SETS

LEONG SIOW HOO

FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2010

MIXTURE MODEL CLUSTERING
FOR VERY LARGE DATA SETS

LEONG SIOW HOO

THESIS SUBMITTED IN FULFILLMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

FACULTY OF SCIENCE
UNIVERSITY OF MALAYA
KUALA LUMPUR

2010

Abstract

This thesis develops mixture model clustering algorithms scalable to data sets that do not fit into the computer memory buffer. Two algorithms, FlexClust and FlexClustMix, are proposed for clustering very large continuous and mixed data sets respectively. The basic framework of the algorithms is to scale down data incrementally using data compression, and incorporates the later compressed data into the current model with the ability to recover new clusters that have been missed out in the earlier compressed data. It consists of three modules: 1) incremental compression, 2) detecting change in cluster structure, and 3) update of current model.

In FlexClust, Gaussian mixture model is used to compress data. The incorporation of the incrementally compressed data into the current model is done through the proposed bi-Gaussian mixture model. In FlexClustMix, a mixture model for mixed data, the Gaussian location model, which speeds up parameters estimation, is proposed for compression. The incorporation of the incrementally compressed data into the current model is done through the proposed bi-Gaussian location model.

A model selection criterion, modified Bayes factor (MBF), is proposed to detect changes in clusters structure due to the incrementally added data and to recover any small clusters that have been missed out in the initial sample.

FlexClust and FlexClustMix are tested over very large continuous and mixed data sets respectively and the results are promising.

Acknowledgement

First and foremost, I would like to express my sincere gratitude and respect to my supervisor Professor Dr Ong Seng Huat for his unreserved guidance, great patience and constant encouragement. He always gives me instructive and prompt advice. I benefited very much from his critical reviews and inspiring comments of the drafts. It is my great pleasure to have been working with him.

My sincere appreciation goes to my friends for walking with me. My heartfelt gratitude goes to my family for encouraging and supporting me all the way. I am indebted to their loves.

Finally, I gratefully acknowledge the scholarship from UiTM that supported my study.

Glossary

Buffer memory/ Buffer
A portion of a computer's memory that is set aside as a temporary holding place for data that is being sent to or received from an external device, such as a hard disk drive, keyboard or printer. It is different from hard disk space.

Classification
Given a set of predefined categorical classes for the target variable, determine to which of these classes a specific unit belongs.

Classification accuracy
The rate of allocate units in the right class, in relation to the given classification. For example, classify cat in the classification of animal is a right allocation. Classification accuracy = 1-misclassification rate. (see misclassification rate).

Complete pass/scan through the data
Read or search all the data.

Condensed/compressed data
A specific set of quantities (prototype) that used to summarize or describe the data points having specific structure particularly a dense region.

Cluster structure
The features of cluster distributions such as orientation, size, and shape.

Data condensation/compression
The process to summarize or describe the data points having specific structure particularly a dense region.

Data points in buffer memory
(see buffer memory).

Hard disk space
Space that available on the hard disk for storing files. It is different from buffer memory.

Incremental data condensation/compression
Data condensation/compression that is being carried out part by part on a data set. (see data condensation/ compression).

Labelled data
Data that the class of each data point is known. (see unlabelled data).

Misclassification
Allocation of unit in a wrong class, in relation to a given classification. For instance, a business is classified in Trade instead of Industry.

Misclassification rate
Rate of misclassification. It is equal to the number of misclassified units divided by the total number of units. (see misclassification)

Mixture distribution
A probability distribution which is expressed as a convex combination of other probability distributions.

Prototype
Summarized information for the condensed/compressed data (see condensed/ compressed data).

Prototype system
A set of prototypes. (see prototype).

Scale down data
A process to reduce the size of a data set under study.

Scalable
Capable of being changed in size. For example, a Web site's design and hardware are considered scalable if the site can handle a significant increase in traffic.

Unlabelled data
Data that the class of each data point is unknown. (see labelled data)

Table of Contents

Abstract ii
Acknowledgement iii
Glossary iv
Table of Contents vi
List of Figures viii
List of Tables x

1 Introduction 1

1.1 Clustering and Data Mining 1
1.1.1 Finite Mixture Model 3
1.2 Challenges in Clustering Very Large Data Sets 10
1.3 Complications in Clustering Analysis 13
1.4 The Thesis 15
1.4.1 Motivation 15
1.4.2 Objectives of the Thesis 15
1.4.3 Contribution of the Thesis 16
1.5 Overview of Thesis 17

2 A Review of Clustering for Massive Data Sets 19

2.1 Scale-Up Model-based Clustering Algorithm 19
2.2 Scale Down Data Techniques 21
2.2.1 Sampling 21
2.2.2 Data Condensation 26
2.3 Change in Clusters Structure 35
2.4 Estimating the Number of Clusters for Condensed Data 37
2.5 Mixed Variables Model-based Clustering 39

3 Determining the Number of Components in Bi-mixture model 42

3.1 Semi Supervised Learning and Partial Classification 42
3.2 Detecting Change in Clusters Structure: A Modified Bayes Factor 45
3.3 Properties of MBF 53
3.4 Bi-Gaussian Mixture Model 53
3.5 Experimental Result 56
3.5.1 Simulation Study 1 57
3.5.2 Simulation Study 2 59
3.6 Conclusion 61

4 Scalable Clustering Algorithm for Very Large Data 62

4.1 Incremental Compression Into Flexible Number of Clusters (FlexClust) ... 62
4.2 The FlexClust Clustering Algorithm ... 63
4.3 Components of the FlexClust Architecture ... 65
4.3.1 Incremental Sampling and Compression ... 65
4.3.2 Model Update ... 66
4.3.3 Model at a Certain Iteration ... 70
4.4 Properties of FlexClust Algorithm ... 70
4.5 Experimental Evaluation ... 71
4.5.1 Simulation Study 1 ... 75
4.5.2 Simulation Study 2 ... 82
4.5.3 Simulation Study 3 ... 84
4.6 Application to Real Data ... 86
4.7 Very Large Simulated Data ... 89
4.8 Discussion and Conclusion ... 90

5 Scalable Mixture Model Clustering Algorithm for Mixed Data 97

5.1 Conditional Gaussian Model ... 97
5.2 Gaussian Location Model (GLM) ... 103
5.3 Simulation for GLM ... 106
5.3.1 Simulation Study 1 ... 106
5.3.2 Simulation Study 2 ... 109
5.3.3 Simulation Study 3 ... 110
5.4 Scalable Clustering Algorithm for Mixed Data ... 112
5.5 Simulation for FlexClustMix ... 113
5.6 Conclusion ... 119

6 Conclusion and Future Work 121

6.1 Contributions ... 121
6.2 Application ... 125
6.3 Limitation and Future work ... 125

References

List of Figures

Figure 3.1. True clusters structure (scatter plot) and the MLE of means and covariances for the model fitted by: a) classified data (‘o’, blue dotted line), unclassified data (‘x’, red dashed line), and bi-Gaussian mixture model (‘+’, black solid line,), and b) bi-Gaussian mixture model (‘+’, black solid line), combined data (‘x’, green dashed line), and true model (‘o’, red solid line). (Note: ellipsoids visualized by 90% normal tolerance). 59

Figure 3.2. True clusters structure (scatter plot) and the MLE of means and covariances for the model fitted by: a) classified data (‘o’, blue dotted line), unclassified data (‘x’, red dashed line), and bi-Gaussian mixture model (‘+’, black solid line), and b) bi-Gaussian mixture model (‘+’, black solid line), and combined data (‘x’, green dashed line). (Note: ellipsoids visualized by 90% normal tolerance). 59

Figure 4.1. The overview of FlexClust. 65

Figure 4.2. Three possibilities when a cluster T_1 is compared to the existing 5 clusters: a) add a new cluster, b) merge with an existing cluster, and c) merge two existing clusters. 67

Figure 4.3. True clusters structure (scatterplot of 10% of the total data points) for the data simulated from a seven-component multivariate normal mixture, which has different shape, volume and orientation. (Note: ‘+’ represent means, and covariances are visualized by 90% normal tolerance ellipsoids). 77

Figure 4.4.True clusters structure (scatterplot of 10% of the total data points) and the MLE of the true model means and covariances (‘x’, red dotted line) compared to the current model (‘+’, black solid line) fitted from incremental compression after: a) initial sample, b) 3 samples, and c) the final sample, in one of the experiments of FlexClust [500]. (Note: covariances are visualized by 90% normal tolerance ellipsoids). 78

Figure 4.5. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) sufficient EM [500]-misses out cluster 6 but identifies superfluous clusters at cluster 1 and 2, and b) sufficient EM [800]-misses out cluster 6 but identifies 2 identical cluster 4. (Refer Figure 4.3 for note.) 78

Figure 4.6. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) strategy III [500], and b) strategy III [1000]. Both miss out cluster 6 but identify superfluous clusters with k=10. (Refer Figure 4.3 for note.) 78

Figure 4.7. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: (a) FlexClust [500] (*k* = 7*), (b) FlexClust [1000] (*k* = 7*). (Refer Figure 4.3 for note.) 81

Figure 4.8. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) sufficient EM [500], and b) sufficient [800]. Non global maximum likelihood estimators and no assignment to some of the clusters. (Refer Figure 4.3 for note.) 81

Figure 4.9. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) strategy III [500], and b) strategy III [1000]. Non global maximum likelihood estimators. No assignment to one of the 10 clusters and some clusters are sparse. (Refer Figure 4.3 for note.) 81

Figure 4.10. Classification structure obtained using: a) SPSS TwoStep (*k* = 5), and b) CLARA (*k* = 5). Plots use 10% of the total observations. 82

Figure 4.11. RGB image of the St Paulia data set 88

Figure 4.12. Ground true image on St Paulia RGB image data in (a). Segmentation results on the image data obtained by: b) FlexClust [1000] (*k* = 24), c) FlexClust [2000] (*k* = 34), d) sufficient EM [1000], (*k* = 26), e) sufficient EM [2000] (*k* = 27), f) strategy III [1000] ((*k* = 11), and g) strategy III [2000] (*k* = 10). 88

Figure 4.13. Difference of mean misclassification rates between FlexClust and sufficient EM. 92

Figure 4.14. Difference of mean misclassification rates between FlexClust and strategy III. 93

Figure 4.15. Percentages of experiments that are able to identify the correct number of clusters and correct clusters 94

Figure 4.16. Performance of average execution time according to (a) number of clusters, (b) data size. 95

Figure 5.1. Plot of the simulated data for (a) sample 1 in 3D using the mixed data, (b) sample 1 using the continuous variables, (c) combined data of sample 1 and 2 in 3D using the mixed data, and (d) combined data of sample 1 and 2 using the continuous variables. (Red dot and black star are two clusters of sample 1; green circle and blue triangle are two clusters of sample 2). 117

List of Tables

Table 1.1. Parameterizations of the covariance matrix Σ_k in the Gaussian mixture models and their geometric interpretation. 5

Table 3.1. Classification accuracy and log-likelihood for bi-Gaussian mixture model and Gaussian mixture model in simulation study 1. The bi-Gaussian mixture model fitted on data that divided into classified and unclassified data according to different ratio. The Gaussian mixture model fitted on the combined data...57

Table 3.2. Results for models obtained from bi-Gaussian mixture model and Gaussian mixture model in simulation study 1. Unclassified data in the bi-Gaussian mixture model are generated from different mixture proportions and with new cluster added. The Gaussian mixture model fitted on the combined data...57

Table 3.3. Table. Classification accuracy and log-likelihood for bi-Gaussian mixture model and Gaussian mixture model in simulation study 2. The bi-Gaussian mixture model fitted on data that divided into classified and unclassified data according to different ratio. The Gaussian mixture model fitted on the combined data...60

Table 3.4. Results for models obtained from bi-Gaussian mixture model and Gaussian mixture model in simulation study 2. Unclassified data in the bi-Gaussian mixture model are generated from different mixture proportions and with new cluster added. The Gaussian mixture model fitted on the combined data...60

Table 4.1. Number of clusters and percentages of getting the correct clusters obtained using sufficient EM, SPSS TwoStep, strategy III, CLARA and FlexClust algorithm on the simulated data. (Numbers in the brackets indicate the number of prototypes for sufficient EM algorithm, and the sample size for Strategy III and the proposed FlexClust algorithm). 76

Table 4.2. Misclassification rate obtained using sufficient EM, SPSS TwoStep, strategy III, CLARA and FlexClust algorithm on the simulated data. (Refer Figure 4.3 for note.) 79

Table 4.3. Misclassification rate and numbers of clusters using sufficient EM, SPSS TwoStep, strategy III, CLARA, and FlexClust on the data generated from *wine* data set. See caption of Table 4.1. 83

Table 4.4. Misclassification rate and numbers of clusters using sufficient EM, SPSS TwoStep, strategy III, CLARA, and FlexClust on the data generated from *iris* data set. See caption of Table 4.1..85

Table 4.5. Number of clusters obtained using the sufficient EM, strategy III and FlexClust on the St Paulia data. See caption of Table 4.1........................89

Table 4.6. Misclassification rates for one-time compression and incremental compression methods for data sets with different dimensions.................91

Table 4.7. Misclassification rates for incremental sampling and random sampling for data sets with different dimensions. 93

Table 5.1. Misclassification rates for simulation study 1 using location model, restricted location model and the proposed Gaussian location model. ...107

Table 5.2. Average estimates and standard errors of the parameters means and covariances for the two continuous variables for simulation study 1 using location model, restricted location model, and the proposed Gaussian location model. 108

Table 5.3. Misclassification rate for simulation study 2 using restricted location model and the proposed Gaussian location model. 110

Table 5.4. The means, covariances of two continuous variables, number of observations for each level of the multinomial variables, for four groups from Franco et al. (2002)..111

Table 5.5. Means, variances, covariances, and number of observations for each level of the multinomial variable for four clusters in simulation study 3 obtained using the modified location model and the proposed model..................111

Table 5.6. Average estimates of means and covariances and misclassification for the bi-Gaussian location model and the Gaussian location model on the simulated data. The bi-Gaussian location model fitted on the simulated data that divided into two samples according to different ratios. The Gaussian location model fitted on the combined data............................115

Table 5.7. Average estimates of means and covariances and misclassification for the bi-Gaussian location model and the Gaussian location model on the simulated data. Data for sample 2 in the bi-Gaussian location model are generated from different mixture proportions and with new cluster added. The Gaussian location model fitted on the combined data.....................115

Table 6.1. Comparison of results for the proposed FlexClust and the other compression-based and sampling-based methods.................................123

Table 6.2. Comparison of results for the proposed Gaussian location model and other parametric models for clustering mixed data. 124

1

Introduction

1.1 Clustering and Data Mining

Clustering is a process used to explore inter relationships among a set of data items (objects, observations, feature, etc.), and group them according to their similarities. Cluster analysis is also known as an unsupervised learning method because there is no priori group labeling for the data items. Generally, there are five clustering methods: partitioning, hierarchical, density-based, grid-based and model-based. The partitioning methods divide a given data set into clusters to optimize an objective function, such as sum of within group sum of squares (Ward, 1963) and distance between groups. Partitioning methods use iterative relocation technique to relocate the data items until there is no change in the gravity centres. The most well known partitioning methods are the *k*-means and *k*-medoids. Some scalable partitioning clustering algorithms have been introduced, for example, CLARA (Kaufman & Rousseeuw, 1990) and scalable *k*-means (Bradley, Fayyad & Reina, 1998a). The disadvantage of this method is that the clusters formed have only spherical shapes. Another widely used clustering method is the hierarchical method, in which data items are grouped into a tree of clusters or dendrogram. There are two types of hierarchical method: agglomerative and divisive. In the agglomerative method, the clusters are formed in a bottom up fashion until all data items belong to the same cluster, whereas the divisive method splits the data items into smaller clusters in a top down fashion until each cluster contains only singleton. Various criteria are used to decide which clusters should be merged or split. The most typical criterion used in the agglomerative method is to merge the closest pair of clusters based on a specified measure such as single-link, complete link and average-link. However, the effectiveness of the splitting process relies on the types of measure used. The hierarchical methods suffer from their inability to perform adjustment once the splitting or merging decision is made. Some examples of scalable hierarchical clustering algorithms are shown in BIRCH (Zhang, Ramakrishnan & Livny, 1996), CURE (Guha, Rastogi & Shim, 1998) and CHAMELEON (Karypis, Han & Kumar, 1999). In density-based methods, clusters are formed based on density of data items in a region, where for each data item of a cluster the neighbourhood of a given radius has to contain at least a minimum number of data items. The most well known density-based clustering algorithm is the DBSCAN (Ester, Kriegel, Sander & Xu, 1996). Some of the works on clustering large data sets using density-based methods are shown in DBCLASD (Xu, Ester, Kriegel & Sander, 1998) and OPTICS (Ankerst, Breunig, Kriegel & Sander, 1999). In grid-based method, the clustering space is first quantized into a finite number of cells, and then cells that contain more than a certain number of points are combined to form clusters. Examples of the scalable grid-based clustering algorithms are STING (Wang, Yang & Muntz, 1997) and CLIQUE (Agrawal, Gehrke, Gunopulos & Raghavan, 1998).

In practice, most clustering is done based on heuristic but intuitively reasonable procedures. Although considerable work has been researched on these methods, their statistical properties are generally unknown, and there is little systematic guidance provided for solving basic practical questions in cluster analysis such as: 1) how many clusters are in the data, 2) which clustering method should be used, and 3) how outlier should be handled (Fraley & Raftery, 1998). Actually, clustering strategies can be based on probabilistic models (Bock, 1996). Using the inferential approach, the conditions for a clustering method to work well can be clarified. This has led to the development of new clustering methods, for instance, finite mixture models have been applied in cluster analysis. Examples of pioneering works can be seen in Day (1969) and Wolfe (1967). This model-based clustering method will be the focus of this thesis.

1.1.1 Finite Mixture Model

In mixture model clustering, the *d*-dimensional random observations of size *n*, $\mathbf{x}_1,...,\mathbf{x}_n$, are assumed to have been generated from a mixture of a finite number, say *G*, of underlying probability distributions in which each component represents a different group or cluster. Let $f_k(\mathbf{x}_i \mid \theta_k)$ be the conditional probability density function for an observation $\mathbf{x}_i$ given that it is from the *k*-th component parameterised by θ_k, the mixture density for each $\mathbf{x}_i$ is expressed as

$$f(\mathbf{x}_i \mid \mathbf{\Psi}) = \sum_{k=1}^{G} \pi_k f_k(\mathbf{x}_i \mid \theta_k), \quad \text{i} = 1, 2, \ldots, n \tag{1.1}$$

where π_k is the non negative mixture proportion for the *k*-th component which satisfies $\sum_{k=1}^{G} \pi_k = 1$, and $\mathbf{\Psi} = (\pi_1, \ldots, \pi_G, \theta_1, \ldots, \theta_G)$ is the vector consists of all the unknown parameters.

Usually, the component-conditional densities are taken to be the same parametric family (Ng & McLachlan, 2003). This thesis concentrates on the case where $f_k(\mathbf{x}_i \mid \theta_k)$ is Gaussian as it is a model that has been used with considerable success in a number of applications (McLachalan & Basford, 1988; Banfield & Raftery, 1993; Celeux & Govert, 1995; Dasgupta & Raftery, 1998). On top of that, density estimation theory states that any distribution can be effectively approximated by Gaussian mixture model (Bradley et al., 1998b). In Gaussian mixture models, the parameter θ_k consists of a mean vector μ_k and a covariance matrix Σ_k, and the density has the form

$$f_k(\mathbf{x}_i \mid \theta_k) = \phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k) = (2\pi)^{-\frac{p}{2}} \mid \Sigma_k \mid^{-\frac{1}{2}} \exp\left\{ \frac{1}{2} (\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k) \right\}. \quad (1.2)$$

The clusters fitted by Gaussian mixture model centred at the means μ_k, and the covariance Σ_k determines their other geometric characteristics like shapes, volumes and orientations.

Banfield et al. (1993) developed a Gaussian model-based clustering framework to allow features of cluster distributions (orientation, size, and shape) to vary between clusters or constrained to be the same for all clusters. This is done by parameterizing the covariance matrix, Σ_k, in term of its eigenvalue decomposition in the form

$$\Sigma_k = \lambda_k D_k A_k D_k^T, \quad (1.3)$$

where D_k is the orthogonal matrix of eigenvectors, A_k is a diagonal matrix with eigenvalues of Σ_k on the diagonal, and λ_k is a scalar. The orientation of the principal components of Σ_k is determined by D_k. The features of cluster distributions are estimated from the data. Table 1.1 shows the geometric interpretation of various parameterizations adopted from Fraley & Raftery (1998, 2003).

The maximum likelihood estimate (MLE) of Ψ based on a set of n independent observations, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, is

$$\hat{\mathbf{\Psi}} = \underset{\mathbf{\Psi}}{\arg\max} \log L(\mathbf{\Psi} \mid \mathbf{x}) \quad (1.4)$$

where $\log L(\Psi \mid \mathbf{x})$ is the log-likelihood function given by

$$\log L(\mathbf{\Psi} \mid \mathbf{x}) = l(\mathbf{\Psi}) = \log \prod_{i=1}^{n} f(\mathbf{x}_i \mid \mathbf{\Psi}) = \sum_{i=1}^{n} \log \sum_{k=1}^{G} \pi_k \phi_k (\mathbf{x}_i ; \theta_k) \quad . \qquad (1.5)$$

In general, the MLE of parameters of mixture model can be estimated iteratively by applying the expectation-maximization (EM) algorithm (Dempster, Laird & Rubin, 1977).

Table 1.1. Parameterizations of the covariance matrix Σ_k in the Gaussian mixture models and their geometric interpretation.

Σ_k	Distribution	Volume	Shape	Orientation	Identifier in MCLUST
λI	Spherical	equal	equal	NA	EII
$\lambda_k I$	Spherical	variable	equal	NA	VII
λA	Diagonal	equal	equal	coordinate axes	EEI
$\lambda_k A$	Diagonal	variable	equal	coordinate axes	VEI
λA_k	Diagonal	equal	variable	coordinate axes	EVI
$\lambda_k A_k$	Diagonal	variable	variable	coordinate axes	VVI
λDAD^T	Ellipsoidal	equal	equal	equal	EEE
$\lambda D_k A D_k^T$	Ellipsoidal	equal	equal	variable	EEV
$\lambda_k D_k A D_k^T$	Ellipsoidal	variable	equal	variable	VEV
$\lambda_k D_k A_k D_k^T$	Ellipsoidal	variable	variable	variable	VVV

EM Algorithms for Clustering

The EM algorithm (Dempster et al., 1977) for clustering is a general approach to maximize likelihood in the presence of a set of unobservable group-indicator $\mathbf{z}_1,...,\mathbf{z}_n$ treated as incomplete data. Each of these indicators has the form $\mathbf{z}_i = (z_{i1},...,z_{iG})$ with

$$z_{ik} = \begin{cases} 1 & \text{if } \mathrm{x}_i \text{ belongs to group } k \\ 0 & \text{otherwise} \end{cases} , \qquad (1.6)$$

and $\mathbf{z}_1,...,\mathbf{z}_n$ are independently and identically distributed according to a multinomial distribution taking G mutually exclusive values with probabilities $\pi_1,...,\pi_G$. Let the "complete data" be $\mathbf{y}_i = (\mathbf{x}_i, \mathbf{z}_i)$, the complete data log likelihood is given by

$$\log L(\mathbf{\Psi}, \mathbf{z} \mid \mathbf{x}) = \sum_{i=1}^{n} \sum_{k=1}^{G} z_{ik} \log[\pi_k \phi_k (\mathbf{x}_i | \theta_k)]. \tag{1.7}$$

The EM algorithm estimates the parameters in the mixture models by iterating between two steps, the E-step and the M-step, repeatedly until the estimates converged.

E-step: Compute the expected value of the complete log-likelihood, conditioned on the observed data $\mathbf{x}_i$ and the current parameter estimates. Since (1.7) is linear with respect to z_{ik}, the E-step is reduced to the computation of the conditional expectation of z_{ik} given the observation $\mathbf{x}_i$ and current parameter estimates. For a given current parameter estimate~~s~~ at *t*-th iteration, $\mathbf{\Psi}^{(t)}$,

$$z_{ik}^{(t)} = \frac{\pi_k^{(t)} \phi_k (\mathbf{x}_i \mid \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{k=1}^{G} \pi_k^{(t)} \phi_k (\mathbf{x}_i \mid \mu_k^{(t)}, \Sigma_k^{(t)})}, \tag{1.8}$$

for $i = 1,\dots,n$ and $k = 1,\dots,G$.

M-step: Maximize the complete data log-likelihood (1.7) with each $z_{ik}^{(t-1)}$ replaced by its current conditional expectation $z_{ik}^{(t)}$. Update the estimates to $\mathbf{\Psi}^{(t+1)}$ as follows

$$n_k^{(t+1)} = \sum_{i=1}^{n} z_{ik}^{(t)} \tag{2.9a}$$

$$\pi_k^{(t+1)} = \frac{n_k^{(t+1)}}{n} \tag{2.9b}$$

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{n} z_{ik}^{(t)} \mathbf{x}_i}{n_k} \tag{2.9c}$$

$$\Sigma_k^{(t+1)} = \frac{1}{\sum_{i=1}^{n} z_{ik}^{(t)}} \left[\sum_{i=1}^{n} z_{ik}^{(t)} (\mathbf{x}_i - \mu_k^{(t+1)})(\mathbf{x}_i - \mu_k^{(t+1)})^T \right] \tag{2.9d}$$

As a greedy algorithm, EM ensures the log-likelihood increases monotonically and satisfies

$$L(\Psi^{(t+1)}) > L(\Psi^{(t)}) . \tag{1.10}$$

The iteration stops when the increase in the log-likelihood becomes smaller than a pre-set threshold. Let the MLE for $\mathbf{\Psi}$ be $\hat{\mathbf{\Psi}}$, the observed data $\mathbf{x}_i$ can be assigned to the component of the mixture with the highest estimated posterior probability where

$$\hat{z}_{ig} = \begin{cases} 1 & \text{if } g = \arg\max_{k} \hat{z}_{ik} \\ 0 & \text{otherwise} \end{cases} . \tag{1.11}$$

Overall, the EM algorithm is an efficient and stable numerical procedure for finding MLEs. However, EM algorithm is sensitive to the initialization and thus it does not guarantee converge to the global maximum. The convergence rate of EM may be very slow if the clusters are not well separated. It can be seen from (1.7) that the E-step scans through each observation $\mathbf{x}_i$. Hence, for large data set that is too huge to be loaded into the buffer memory, the EM algorithm would be impractical.

Bayesian Model Selection Criteria

One of the advantages of using mixture model clustering is that it allows the use of approximated Bayes factors to compare models. Through the available approaches, for instance EM algorithm, a sequence of mixture models with different parameter estimates is obtained for a range of values of the number of clusters. Most often, the selection of the best fitted model among all includes maximization of likelihood.

Let $\mathbf{x} = \mathbf{x}_1, ..., \mathbf{x}_n$, be d-dimensional random observations of size n to be described using a model M_k selected from a sequence of candidate models M_1, …, M_L, which are not necessary nested. Assumed that each M_k is uniquely parameterised by a vector $\mathbf{\Theta}^{\mathrm{k}}$, and let p denotes the functionally independent parameters to be estimated in $\mathbf{\Theta}^{\mathrm{k}}$, and the number of clusters for the model, k, is not considered as an independent parameter. Let $L(\mathbf{\Theta}^k \mid \mathbf{x})$ represents the likelihood of the data for the model M_k. If EM-based approaches are used to find the maximum mixture likelihood, an approximation to twice the log Bayes factor called the Bayesian Information Criterion (BIC) (Schwarz, 1978) can be obtained which defined as

$$\mathrm{BIC} = -2\log L(\hat{\mathbf{\Theta}}^k \mid \mathbf{x}) + p\log n \qquad (1.12)$$

where $\log L(\hat{\mathbf{\Theta}}^k \mid \mathbf{x})$ is the maximized mixture log-likelihood for model.

For model with more clusters, the number of parameters also increases, and this leads to increase in the likelihood. Hence, the likelihood cannot be used directly in assessing the best fitted mixture model. An additional term is added to the log-likelihood in BIC to penalize the complexity of the model so that more parsimonious model with less number of clusters could be selected as the best fitted model. From (1.12), the larger the $\log L(\hat{\mathbf{\Theta}}^k \mid \mathbf{x})$, the more likely the model is better fitted, and this implies that the smaller value of BIC suggests stronger evidence for the model. The BIC can be used to compare models with differing parameterisations, differing numbers of components, or both. Although regularity conditions for the BIC do not hold for mixture models, there is considerable theoretical and practical support for its use. Leroux (1992) showed that model selection based on BIC values does not underestimate the number of groups asymptotically. Keribin (1998) showed that BIC is consistent for the number of groups. A range of applications using BIC for model selection has shown good results (Dasgupta & Raftery, 1998; Mukerjee, Feigelson, Babu, Murtagh, Fraley & Raftery, 1998, Stanford & Raftery, 2000).

Advantages of Mixture Model Clustering

It has been recognized that mixture models are mathematically sound to answer the practical questions arising in cluster analysis (McLachlan & Basford, 1988; Fraley & Raftery, 1998). First, the question of how many clusters are in the data can be approached through an assessment of how many components are needed in the mixture model, and models with different number of components can be compared. Second, the problem of choosing an appropriate clustering method can be recast as statistical model choice. Thirdly, the outliers can be handled by adding one or more components in the mixture model. Mixture models have since been increasingly used for clustering. However, their practical applications without modification can be inadequate for large data sets.

Clustering Moderately Large Data Sets

Mixture model clustering methods, which originated from statistics, have been developed to identify natural groups for relatively small data sets, making them inappropriate if not infeasible for massive data sets. This complication challenges the emergence of powerful mixture model clustering algorithms to manage the data avalanche. Over the years, mixture model clustering algorithms for moderately large data sets have received growing interest, and the problem has remained an extensive research topic. For instance, some scalable mixture model clustering algorithms have been developed and applied in the area of microarray analysis (McLachlan, Bean & Ng, 2008), magnetic resonance image segmentation (Banfield & Raftery, 1993; Wehren, Simonetti & Buydens, 2002; Wehrens, Buydens, Fraley & Raftery, 2004), software metrics and tomography (Maitra, 2001), web mining (Steiner & Hudec, 2007). However, advances in computing technology make data collection and storage easier than ever before and this has led to a rapid growth of vast databases. The existing mixture model clustering algorithms developed for moderately large data sets cannot cop with the vast databases. To find patterns in very large data sets becomes even more challenging than ever. This will be discussed in the following section.

1.2 Challenges in Clustering Very Large Data Sets

As data mining is characterized by intensive computations on large amount of data, the development of new clustering analysis algorithms assist in transforming the data into valuable knowledge encounters a few complications discussed below.

Scalability and Exhaustiveness

Massive data set may involve several millions of records. The most significant challenges in clustering massive data set are scalability and consistent quality performance as the data size grows. According to Bradley, Fayyad and Reina (1998a, 1998b, 2000) and Fraley (1999), a clustering algorithm is considered scalable if the running time grows linearly in proportion to the size of the database and can be handled by the main memory and disk space. Apparently, the ability of clustering algorithms to perform well with massive data is constrained by three main resources: size of database, memory space and time. In traditional applications of clustering in data mining, database size tends to be the dominant problem. However, in many real applications, the bottleneck is time and memory. When data is typically in oversupply, unfortunately, the ability of the current clustering algorithms to analyze these data sets within a reasonable time frame and within the available computational resources have not kept pace. As a result, full advantage of the data cannot be taken of because most of the data go unused and the obtained model may under fit. Thus, the development of clustering algorithms scalable to data size and memory capacity becomes a priority. Currently, the development of efficient algorithms concentrates on the feasibility of mining data sets that do not fit in main memory but require sequential scans of the data from the disk. Some related works have been shown by Shafer, Agrawal and Mehta (1996) and Bradley et al. (1998a, 1998b, 2000).

With the expansion of the Internet, data arrives at an explosive rate. It becomes a reality that the present clustering algorithms cannot even mine a fraction of the massive data within satisfactory time. In particular, data accumulates faster than it can be mined. In this situation the clustering algorithms are not exhaustive, and some data remains unexplored and its amount grows tremendously as time progresses. This leads to more storage of data needed to be preserved for future use and the transfer of data from one storage device to another during clustering processing is difficult.

Open data stream

When the source of data is an open data stream, the application of clustering algorithms based on the concept of mining a fixed size data set becomes questionable and determining how many clusters in the data set becomes a key challenge. Most clustering algorithms for data mining assume that training data is a random sample drawn from a stationary distribution. These clustering algorithms learn an incorrect number of clusters and cluster structure when they erroneously assume that the database is stationary but in fact it is changing. For an open data stream, the number and structure of clusters may change over time. In many cases, it is more accurate to assume that there are possible changes in the existing model structure where new clusters are added to it or the existing clusters are merged. Thus, it is challenging to develop clustering algorithms in data mining that operate continuously and dynamically within strict resource constraints, incorporate data sets as they arrive and keep the learned model up to date so that it will never lose potentially valuable information.

High dimensional data

Apart from the large number of data points, another aspect of massive data is the high dimensionality (Fayyad & Smith, 1996). The ability of present clustering method to handle high dimensional data is still very limited. There are two major challenges for clustering high dimensional data. The first one is the curse of dimensionality (Bellman, 1961). With increasing dimensionality, the time complexity of many existing clustering algorithms to explore clusters is exponential and soon become computationally intractable and therefore inapplicable in many real applications. Secondly, the specificity of similarities between points in a high dimensional space diminishes. It was proven in Beyer, Goldstein, Ramakrishnan and Shaft (1999) that for any point in a high dimensional space, the expected gap between the Euclidean distance to the closest neighbour and that to the farthest point shrinks as the dimensionality grows. This phenomenon may render many clustering algorithms for data mining ineffective and fragile because the model becomes vulnerable to the presence of noise. Most of the efforts to tackle high dimensional data clustering are confined to conducting pre-clustering step either by feature extraction or dimensionality reduction. However, both pre-clustering methods have their shortcomings.

This thesis focuses on the first two challenges in clustering very large data sets mentioned above. Scalable model-based clustering algorithms are proposed to work within memory limit, and can be applied for clustering open data stream. Effort to tackle high dimensional data is beyond the scope of this thesis. It is assumed that the data sets to be used in this thesis are either of reasonable dimension or appropriate techniques have been employed to reduce their dimensions.

1.3 Complications in Clustering Analysis

When the central issues for clustering large data are focus on computational scalability in term of the data size, dimensionality and memory space, there are special requirements posed by the applications of clustering analysis that complicate the challenges in clustering large data. The following are some of these typical requirements of clustering analysis listed down by Han and Kaufmann (2001) and the related problems that are of concern in this thesis.

> **Ability to deal with attributes of different types**: Most clustering algorithms assume data sets where objects are defined on either numerical values or categorical attributes. In such case, algorithms for clustering continuous data and categorical data can be used respectively. However, the data in the real world usually contains continuous and categorical attributes, that is, mixed attributes. This complicates the situation, in particular, when the similarity between categorical attributes is not taken into consideration during clustering. To solve the problem, most clustering algorithms use the approach of converting one type of the attributes to the other and then apply single-type attribute clustering algorithms. However, such transformation leads to two

major problems: the loss of semantics and waste of storage when the domain of the categorical attribute is large.

Discovery of arbitrary shaped clusters: Different types of clustering algorithms will find different types of cluster structures. Some examples can be seen in Gionis, Mannila and Tsaparas (2007), and Halkidi and Vazirgiannis (2008). Distance-based clustering algorithms using Euclidean or Manhattan distance measures tend to identify spherical clusters with similar size and density.

Minimal requirement for domain knowledge: Many clustering algorithms require some user-defined parameters, such as the number of clusters, average dimensionality of the cluster and etc. However, in practical application, such information is unknown and difficult to determine.

Insensitivity to the order of input records: Some clustering algorithms are sensitive to the input order of the data. For instance, clustering algorithms based on CF-tree: BIRCH (Zhang, Ramakrishnan & Livny, 1996) and TwoStep SPSS (SPSS Inc., 2003) may produce dramatically different clustering result when different input orders are used even though the data set are actually the same.

The proposed clustering algorithms in this thesis apply model-based clustering method to address the above complications.

1.4 The Thesis

1.4.1 Motivation

How well a model-based algorithm performs in clustering massive data is constrained by the size of database, memory space and time spent. Working on compressed data rather than individual data points relaxes all these restrictions at the same time. However, the present compression techniques do not preserve the clusters structure well and cause loss of information. Most practical compression methods are based largely on heuristic but intuitively reasonable procedures. It motivates this thesis to use mixture model for data compression. Mixture model has solid mathematical foundations from the statistics community. It can describe clusters with a variety of shapes from data set either with continuous or mixed attributes, and have the advantage of automatically determining the number of clusters.

To work beyond the limitation of memory buffer, clustering algorithms that compress massive data set incrementally and incorporate the compressed information into the in-memory model are much desired. However, related works only considered the case where the clusters are not overlapping and the number of clusters is assumed known and constant throughout incremental compression. In actual fact, any newly arrived data may cause the in-memory model obsolete. It motivates this thesis to develop scalable clustering algorithm that allows incorporation of newly compressed data into the current model with the flexibility of allowing change in the clusters structure, and operates continuously and dynamically within strict memory constraint.

1.4.2 Objectives of the Thesis

The main objective of this thesis is to develop scalable and exhaustive mixture model clustering algorithms for very large data sets that do not fit in the computer memory buffer. Two algorithms, FlexClust and FlexClustMix are developed respectively for continuous data and mixed data. The proposed algorithms compress data incrementally according to the available memory buffer using the appropriate mixture model according to the type of data and incorporate the compressed information into the current model with the flexibility of allowing changes in the clusters structure and the number of clusters.

The sub-objective of this thesis is to develop a model selection criterion based on the Bayesian approach to determine the changes in clusters structure and number of clusters between the current model and the data drawn incrementally in the memory buffer.

Another sub-objective of this thesis is to develop models according to the type of data for incorporating incrementally compressed information characterized by maximum likelihood estimates (MLEs) into the current models with the flexibility of allowing changes in the clusters structure and the number of clusters.

The last objective of this thesis is to develop a mixture model for clustering mixed data, which reduces the number of parameters to be estimated in the EM algorithm and can be incorporated into the proposed FlexClust algorithm for the development of the FlexClustMix algorithm.

1.4.3 Contribution of the Thesis

The contributions of this thesis are given below.

1 It introduces two scalable mixture model clustering algorithms, FlexClust and FlexClustMix, respectively for continuous and mixed data sets that do not fit in the memory buffer. The algorithms can be applied to find useful

information in the areas that involve large data set such as web mining, image segmentation, software metrics and tomography, and transaction data.

2 It adapts the joint probability distribution for mixed variables that suggested by Cox (1972), which has not been explored any further, to propose a mixture model for mixed data, Gaussian location model. The Gaussian location model reduces the number of parameters to be estimated during the EM algorithm, thus speeding up parameters estimation, and therefore it is suited for clustering very large mixed data sets.

3 It proposes two mixture distributions, the bi-Gaussian mixture model and bi-Gaussian location model, which allow changes in the number of mixture components, to incorporate the incrementally compressed continuous and mixed data respectively into the current models. The models accommodate clusters that recovered from the later compressed data.

1.5 Overview of Thesis

This thesis consists of six chapters. The present chapter gives the research background and lists the main challenges and complications in clustering very large data sets.

Chapter 2 surveys the techniques used to handle large data set in clustering. Techniques on scaling up model based clustering method and techniques for scaling down data set are discussed. Sequential clustering algorithms that allow change of clusters structure are covered. Clustering methods for mixed data, particularly on conditional Gaussian model and its derivations, are reviewed.

Chapter 3 extends the work of partial classification to the case where the unclassified data is drawn from or outside the population of the trained model. It paves the way to the development of the bi-Gaussian mixture model. The derivation of a proposed model selection criterion, modified Bayes factor (MBF), is presented.

In Chapter 4, the framework used to develop a scalable mixture model clustering algorithm FlexClust for continuous data sets that do not fit in the memory is presented. This framework consists of three parts: incremental data compression, determination of cluster identity, and incorporation of newly condensed data. The performance of FlexClust is examined for both synthetic and real world data, and the results are compared to a number of current clustering algorithms.

In Chapter 5, a mixture model for clustering mixed data, Gaussian location model, is first developed and its performance is compared to some existing models. The incorporation of the Gaussian location model into FlexClust algorithm to develop a scalable mixture model for clustering very large mixed data sets, FlexClustMix, is described. The evaluation of the FlexClustMix algorithm on very large mixed data is presented.

Chapter 6 concludes and discusses the contributions and limitations of the thesis and presents some research directions for future work.

2

A Review of Clustering for Massive Data Sets

In this chapter, the review starts with the related work in scaling clustering algorithm for large data set in the direction of scaling up algorithm and scaling down data set. The focus is particularly on the method of scaling down data especially incremental data condensation which actually motivates the proposed scalable clustering algorithm. Sequential clustering algorithms that allow change in clusters structure are also reviewed to shed some light in developing the proposed clustering algorithms. This follows the relevant work in estimating the number of clusters specifically for condensed data. Lastly, the clustering algorithms for mixed data are reviewed.

2.1 Scale-Up Model-based Clustering Algorithm

In model-based clustering method, the Expectation-Maximization (EM) algorithm (Dempster, Laird & Rubin, 1977) is usually used for iterative maximum likelihood estimation. McLachlan and Krishnan (1997) had detailed some desired properties for the EM algorithm. Unfortunately, the EM algorithm needs a complete pass through the data in every iteration and this causes very slow convergence. This problem becomes more severe when the data set is large. In most of the research efforts, the direction to scale up model-based methods for large data set clustering is to speed up the EM algorithm convergence.

In Gaussian mixture model where the computational of M-step is simple, the computation time for EM algorithm to convergence depends mainly on the E-step because E-step passes each data points (Ng & McLachlan, 2003). Thus, as an adaptation to scale up EM algorithm to large data sets, various improvements have been done on the E-step. Variants of EM algorithm based on a partial E-step have been proposed to accelerate the convergence rate. For instance, in incremental EM (Neal & Hinton, 1998), E-step is performed to update parameters in blocks of observations at a time before the next M-step is undertaken. The argument is that updating blocks of observations is quicker than a complete scan of all the observations, however, a full E-step is performed for the first scan to avoid premature component starvation (Thiesson, Meek & Heckerman, 2001). For lazy EM (McLachlan & Peel, 2000), the posterior probabilities of component membership of some observations are not updated on every E-step. On the E-step, those observations with maximum posterior probability above a threshold close to 1 are held fixed whereas the remaining observations are updated. In sparse EM (Neal & Hinton, 1998), for a given observation only posterior probabilities of component membership above a certain threshold are updated on E-step while the posterior probabilities of the remaining components of the mixture are held fixed. In both sparse EM and lazy EM, the M-step only update the corresponding mixture parameters, and a full E-step is performed periodically to ensure convergence.

There are also techniques being introduced to switch alternatively with the iteration of EM algorithm so that the convergence of EM can be speeded up. For instances, the conjugate-gradient acceleration of Jamshidian and Jennrich (1993), and Newton approximations, which includes the quasi-Newton methods (Lange, 1995; Aitken & Aitken, 1996; Jamshidian & Jennrich, 1997). However, these techniques were developed for small data sets.

The efforts of scaling up model-based clustering algorithms by speeding up the EM algorithm have lagged behind the development of scalable clustering. Even though these methods reduce computational time, they still require full scan of all the observations in some EM iterations. Apparently, when the data set is too large to be loaded into the memory buffer, these techniques would be impractical. Hence, to date, scaling up model-based clustering algorithms is still not a satisfying solution for very large data sets. A more promising approach would be the scaling down of the data.

2.2 Scale Down Data Techniques

Clustering methods are developed based on small data set. Various adaptations have been applied to scale up these basic clustering methods to accommodate huge data set. Unfortunately, working along this line is to complicate the complexity problem. Alternatively, if the large data set can be scaled down without much loss of information, it can be clustered using the existing clustering methods which perform well on small data set. This is not only assuring the clustering result, but also saves a lot of computational time and enables clustering of large data set works within the limited memory buffer. The critical issue in scaling down data would be how to avoid the loss of information. In general, there are two common approaches in scaling down data in the discussion as follows.

2.2.1 Sampling

A general intuitive approach to scale down data is through sampling. A sample is assumed to reflect the whole data structure and thus the clustering result can be generalized to the entire data. Nevertheless, a number of critical issues have to be addressed: sampling method to be employed, incorporation of sampling in large data clustering, and the appropriate sample size used.

2.2.1.1 Random Sample Training

Researchers usually obtain random sample to represent large data set but perform according to different procedures to speed up large data clustering. Commonly, a random sample is used as training data to obtain a model, and this model from the sample data is then used to perform discriminant analysis to classify the rest of the data. The notion is shown in CLARA (Clustering LARge Application) (Kaufman & Rousseeuw, 1990). CLARA draws a random sample of the data set and applies *k*-medoids to cluster the sample, then classifies the rest of the data points to the nearest medoids of the sample. The procedures are repeated for a few times to find a set of medoids that gives the best clustering result with the smallest average distance. CLARA assumes the number of clusters is known. The procedure with the flavour of discriminant analysis is also applied in model-based clustering for segmenting magnetic resonance image (Banfield et al., 1993; Wehren, Simonetti & Buydens, 2002). A random sample is first taken as training set and fitted into a mixture model. The resulting model is then applied to perform discriminant analysis to classify the remaining data. Basically, it is an extension of the final E-step from the sample model to obtain conditional probabilities of the remaining data (Fradley & Raftery, 2002). Wehrens, Buydens, Fraley and Raftery (2004) showed that this simple method may lead to unstable results. Thus, they suggest two modifications to a stable method with better performance: 1) several tentative models are identified from the sample instead of one, and 2) more EM iterations are used instead of just one E-step to classify the whole data set.

Rocke and Dai (2003) proposed the concept of sampling and subsampling to improve the training of model from sample, and the trained model is used to classify the remainder. In the proposed strategy, a random sample is first fitted using EM algorithm to find the group labels. Multiple stratified samples from the sample are then drawn to find the estimates with the highest likelihood via EM algorithm. The maximum likelihood estimates of the subsample are used as initial values to perform EM algorithm to fit a model for the sample or on a supersample which is much larger than the sample. Multiple random samples are selected to repeat the process of model training. The best model obtained is then used to classify the whole data set. The idea of subsampling the sample of training data is also shown in Davidson and Satyanarayana (2003) to speed up *k*-means clustering. The subsamples of a randomly selected training data are clustered by *k*-means, and then the resulting cluster centers are bootstrapped and averaged to build a single model to be applied to the whole data set.

2.2.1.2 Recovery of Small clusters

Although various strategies as described above have been proposed to use random sample to train a model that can be extended to classify a large population, the question of how accurate is the trained model has not been addressed. The efficiency of the trained model depends on the strategies used to train it, and also the quality of the drawn sample used for training. These training strategies may reduce the accuracy of the classifier model for pathological or non-representative sample. Even in the case where all the clusters in the data are of equal importance, sampling may provide significant different solution from the one obtained from the entire data set (Posse, 2001). Another possible reason which causes break down of the model trained from random sample is when there are small clusters in the large data set. The fact is that random sampling may not be able to capture enough representatives from small clusters, and in some cases the representatives from the small clusters are identified as noise or outlier. As a result, small clusters may be underestimated or being missed out (Gordon, 1986; Fayyad & Smyth, 1996). Fayyad et al. (1996) improved this by proposing iterated sampling in which data points from the remainder that do not fit well (with low probabilities) in any clusters of the model trained from the sample are accumulated for further investigation. If the number of poorly classified data points is small, a small sample from the data set is selected and mixed with these data points to form a second sample. Then, clustering is performed on the second sample to look for tight clusters which are considered as candidates of newly discovered small clusters. Fradley and Raftery (2002) modified the above iterating sampling in the selection of second sample. Instead of merely a small sample from the data set, an equal proportion stratified sample from those which have been fitted well in any clusters is mixed to the poorly classified data points. Fraley, Raftery and Wehrens (2005) extended the iterating sampling to incremental model-based clustering for identifying small clusters in large data set. The poorly classified data points are grouped in a separated cluster, and run one or more steps of EM algorithm. If it improves BIC, the decision of adding new small cluster to the model is made. The procedure repeats and new small clusters can be added incrementally. These latest development in iterating sampling are contradicted with the concept of sample trimming. The basic idea of sample trimming is to remove outliers that are not sufficiently representative of the data set as a whole so that the trimmed sample reflects more accurately the parent population and improves estimates from random sample (Miller, 1986). To enhance the possibility of the inclusion of small clusters at each stage of the incremental sampling in a multistage clustering algorithm, Maitra (2001) proposed a scheme to progressively increase their weights. The progressive scheme of Maitra (2001) and iterated sampling of

Fayyad et al (1996) are adapted in this thesis to propose a procedure to discover small clusters.

To reduce the biases caused by random sampling when the cluster sizes are skewed, Palmer and Faloutsos (2000) proposed density biased sampling which is based on the concept of weighted sample. The sampling method favours clusters containing fewer data points, and thus avoids the missing of these small clusters. However, the implementation of the sampling method needs a prior knowledge of how the data is distributed. Although this can be approximated using hashing based bin labels, still a full database scan is required and it may be computationally intractable for large data set.

2.2.1.3 Optimal Sample Size

While generating good and representative samples remains a challenging issue for sampling, determining an optimal sample size for model training is another difficulty. Domingos and Hulten (2001) used Hoeffding bound (Hoeffding, 1967) to derive a bound on the accuracy of the output as a function of sample size to choose a sample size for *k*-means clustering. A usual way of optimal sample size determination is usually assisted by a learning curve which is constructed from growing samples to bigger and bigger sizes until a stopping criterion is satisfied. Dynamic sampling (John & Langley, 1996) and progressive sampling (Provost, Jensen & Oates, 1999) have been used to draw a sequence of data sets and are stopped when the accuracy of the current data set meets the predefined thresholds. However, these two sampling methods do not consider the computation costs. Meek, Theiesson and Heckerman (2001) incorporated the learning curve method into the model-based clustering method to determine sample size. In a mixture model with known number of components, the learning algorithm grows the sample sizes until the expected costs outweigh the expected benefit associated with training. Actually, it is likely that the sample size required to estimate the model accuracy of the whole data set varies according to data set. Instead of finding the optimal sample size, Fraley et al. (2005) suggested to draw multiple samples in any sampling-based strategy to increase the chances to find a good model for the data. Some related works are shown in Kaufman et al. (1990), Rocke et al. (2003) and Wehrens et al. (2004).

2.2.1.4 Sampling for Memory Buffer

Despite the improvement in clustering speed, applying sampling to scale down data in most of the works is limited to sample model training and extending the trained sample model to the full data set. Bradley et al. (1998a; 1998b; 2000) had shown another branch of using sample to scale down data. They considered a very practical case where the data size is larger than the available memory size, and applied sampling to load data incrementally and incorporated in an incremental clustering algorithm. Indeed, this approach motivates the scalable and exhaustive clustering algorithm in this thesis.

2.2.2 Data Condensation

Data condensation or data compression has been widely applied in image processing which is also known as vector quantization (Gersho & Gray, 1992). The notion of data condensation was initialled to free more disk memory for the computation of large database. To scale down data through data condensation is often referred to as summarizing or describing the data points having specific structure particularly a dense region by a specific set of quantities (prototype). In this case, data condensation can actually be considered as a clustering process at pre-clustering stage. Prototype characterized by triple sufficient statistics has been widely used because it can be accommodated in most of the clustering methods. For instance, a triple of summarized information (*SUM*, *SUMSQ*, *N*), where *SUM* and *SUMSQ* are the linear sum and the sum of square of the compressed data points, and *N* is the number of data points in the compressed set, has been used in Bradley et al. (1998a, 1998b, 2000), Zhang et al. (1996), and Jin, Leung, Wong and Xu (2005), whereas a triple of sufficient statistics $(\bar{x}, S, n)$, where $\bar{x}$ and *S* are the mean and covariance of the compressed data points, and *n* is the number of data points of the compressed set, has been used in Steiner and Hudec (2007) and Tantrum Murua, and Stuetzle. (2002). A set of these prototypes forms a prototype system that represents the original data set, which needs less storage compared to retaining all the data points. However, scaling down data using data condensation faces two critical challenges: 1) condensing data sets that do not fit in the computer memory buffer, and 2) loss of information. These two challenges are closely related to the data condensation procedure, the condensation method and the resulting prototype. The related works in these issues as discussed in the following have inspired the direction of this thesis.

2.2.2.1 Condensation Procedures

In general, there are two procedures for data condensation: 1) one time condensation and independent from the clustering algorithm, and 2) incremental condensation and closely related to the current fit of the clustering model.

One Time Condensation

In the one time condensation procedure, all the data are loaded in the memory, and data condensation is carried out to create in-memory summary of data before any clustering algorithm is performed. One major advantage of the one time condensation procedure is that the condensation is done on the entire data set as a whole and thus avoids loss of information due to partition or sampling of data. There are few well known clustering algorithms in the literature applied data condensation approaches according to this contemplation. BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies) (Zhang et al., 1996) introduces two core concepts for incremental one time condensation: Clustering Feature (CF) and CF tree. The Clustering Feature is a triple summarized information about the compressed compact data points in a subcluster, CF = (N, LS, SS), where N is the number of data points in the subcluster, LS and SS are the linear sum and the square sum of the N data points respectively. The CF tree is a summary of the subclusters which is used for guiding new insertion of subclusters represented by their CF vectors into the closest leaf node. When CFs are incrementally computed, the CF tree will be built dynamically. By controlling both the branching factor and threshold parameters, the CF tree can be rebuilt by splitting or merging the leaf node or non leaf node to prevent it running out of memory. Then, a hierarchical agglomerative clustering algorithm is applied to cluster the leaf nodes of the CF tree. Lastly, k-means is used for refining the clustering of data points and this allows data points grouped in the initial cluster to be separated from that group. This refining step overcomes the inability of pure hierarchical method to perform adjustment once a merge or a split decision has been executed. However, like other tree-structure clustering methods, BIRCH also faces the drawback of dependence of data input order. Based on the framework of BIRCH, a CF for mixed data was developed in the format CF = (*N*, *LS*, *SS*, N_{kl}), where *N*, *LS* and *SS* have the same definitions as BIRCH, and N_{kl} is the number of data points whose *k*-th categorical attribute takes the *l*-level (Chiu, Fang, Chen, Wang & Jeris, 2001). Since CF uses notion of radius and diameter from centroid to compress the dense region, BIRCH and other clustering algorithms that applied CF do not perform well if the original clusters are not in spherical shape (Palmer & Faloutsos, 2000; Han et al., 2001). Apart from that, using CF vector to represent condensed data raises three problems: 1) structural distortions, 2) size distortions, and 3) lost objects (Breunig, Kriegel, Kroger & Sander, 2001). The latter two problems have a rather straightforward solution by weighing each CF but improve very little on the clustering quality. To solve the structural distortion problem, a new prototype for the compressed data points so called Data Bubbles (Breunig et al., 2001) is introduced to speed up the hierarchical clustering algorithm OPTICS (Ordering Points To Identify the Clustering Structure) (Ankerst, Breunig, Kriegel & Sander, 1999). Data Bubbles has the distance measure *B* = (*rep*, *n*, *extent*, *nnDist*), where *rep* is a representative for a set of data and the natural choice is the mean, *n* is the number of compressed data points, *extent* is a radius extends around *rep* such that most of the compressed points are located, and *nnDist* is a function denoting the estimated average *k*-nearest neighbor distances within the compressed data points for some values *k*.

To speed up the time consuming iterative refinement clustering algorithm for very large database, the concept of data condensation has also been incorporated with EM algorithm in Gaussian mixture modeling to develop new algorithms such like EMACF (EM Algorithm for Clustering Features) (Jin et al., 2005) and sufficient EM algorithm (Steiner et al., 2007). The concept in EMACF and sufficient EM algorithms is to condense data points in subclusters from a large data set into prototypes, and then fit a mixture model for the prototype system by a variant of EM algorithm. EMACF algorithm introduces BIRCH-based and grid-based method to condense subcluster and represents it by CF vector whereas sufficient EM employs *k*-means for data compression and represents the condensed data by a prototype characterized by a triple of sufficient statistics $(\bar{x}, S, n)$, where $\bar{x}$ and S are the mean and variance of the subcluster, and n is the number of data points in the subcluster. Instead of the observed data points, the subcluster means are used in the variant of EM in EMACF and sufficient EM. The density of a subculster mean is defined by a mixture of Gaussian distribution weighted by the number of observed data points in the subcluster. The parameters of component means are estimated by the sum of weighted subcluster means. The distinct difference between EMACF and sufficient EM is the estimation of component covariance matrices. The former uses sum of weighted covariance between subcluster means whereas the latter uses not only weighted covariance between subcluster means but also includes the sum of weighted covariance within subcluster means. The advantage of such inclusion is that it takes into account the structure information of the original data, but it results in non monotonically convergence of likelihood function. Thus, additional conditions are considered for termination for sufficient EM, which are the sums of absolute changes in the mixture parameters respectively must be smaller than a set of thresholds (Steiner et al., 2007).

The main challenge in the one time for all data condensation is the availability of memory buffer to load the whole data set at once. With no prior knowledge about a very large database, the issue of how many prototypes should be chosen in the one time condensation to preserve the original model structure seems unclear.

Incremental Condensation

In contrast to the one time condensation that is done a step independent of the clustering algorithm, incremental condensation is done closely related to the current fit of the mixture model. For data mining applications, when the data size is larger than the available memory space, incremental data condensation according to the available memory buffer and incorporating the compressed information into the current model fit is one of the solutions for this problem. The incremental condensation procedure maintains only the prototype system in the memory and purges the data points to free some memory for filling new data points to the memory buffer, and this makes it scalable to very large data sets. On top of that, it is closely related to the current fit of the clustering model and thus provides usable model at any time. Related works are shown in scalable *k*-means (Bradley et al., 1998a) and scalable EM (Bradley et al., 1998b, 2000) algorithms. The two scalable algorithms operate over a single scan of the data and condense the data to retain only the summarized information in the memory buffer, and the data points are purged. The freed memory allows subsequent scans of additional data. Both scalable *k*-means and scalable EM algorithm are similar except: 1) the model parameters are initialized based on *k*-means and Gaussian mixture model respectively, and 2) the current model parameters are updated through extended *k*-means and extended EM (ExEM) respectively. In the two aforementioned algorithms, the data points are scanned once and divided into three sets: 1) a discard set, *DS* (data points which membership are certain), 2) a compressed set, *CS* (data points which are know to belong together), and 3) a retained set, *R*S (data points which membership are uncertain). Compressions are carried out for the data points in the discard set and compressed set. In primary compression, the *DS*s are identified by thresholding the Mahalanobis radius near the mean of clusters of the in-memory model and then they are condensed and summarized within the radius. The prototype of a *DS* is a triple of summarized information (*SUM*, *SUMSQ*, *N*), where *SUM* and *SUMSQ* are the linear sum and the sum of square of the data points in the discard set respectively, and *N* is the number of data points in the discard set. In secondary compression, *k*-means is used to determine the *CS* or the dense and tight subclusters that are not compressed in primary phase, and the type of prototype same as primary compression is formed. These resulting subclusters from the *CS* are merged with each other and existing clusters using hierarchical agglomerative clustering. The current model update is done over the singleton data points from *RS* and the previous model parameters using extended *k*-means and extended EM respectively for scalable *k*-means and scalable EM algorithms. Subsequent refinement of model update is then based on the summarized information from both primary and secondary compressions. The notion of one scan and incremental data condensation enable the algorithms to speed up the clustering process and scale to very large database. However, scalable *k*-means and scalable EM face two main challenges: 1) missing out small clusters due to the incrementally drawn samples which are not representative, and 2) the number of clusters is assumed known and unchanged when new data is incorporated into the current model, which in actual fact may be erroneous. Furthermore, primary compression near the mean of the clusters using Mahalononis radius may overlook any possible nested clusters. Steiner et al. (2007) pointed out that the high degree of data compression into a known number of mixture components in scalable EM limits promising result only to well separated mixture components. Apart from that, ExEM is derived in a heuristic way and it is not easy to ascertain its convergence (Jin et al., 2005). This thesis proposes a new algorithm to overcome the problems in the scalable EM.

2.2.2.2 Condensation Methods

As data condensation is to identify and cluster data points in dense region and then represents them by summarized information, it can be undertaken by any clustering procedure. The popular *k*-means clustering algorithm has been considered widely for data condensation in view of its simplicity to apply and its summarized information can be accommodated in most of the clustering methods. However, using *k*-means for data condensation, for instances, BIRCH, scalable k-means, scalable EM, EMACF and sufficient EM algorithms may destroy the original model structure if the clusters are not homogeneous and spherically shaped, and not reasonably well separated. Ordonez and Omiecinski (2002) suggested a model-based clustering algorithm, FREM (Fast and Robust EM), using EM algorithm itself to condense data points according to a desired number of components. FREM condenses data points from the same mixture component during the E-step and represents them by the prototypes similar to scalable EM, i.e. (*SUM*, *SUMSQ*, *N*). In the M-step, these prototypes are used to update the mixture parameters in order to avoid repeated scans over the data points. Apparently, FREM speeds up the computation of the M-step but overall the algorithm is still considered slow for large data clustering because the computational time spent in M-step depends only on the number of groups in the mixture models (Ng et al., 2003). Tantrum, Murua and Stuetzle (2002) employed hierarchical model-based clustering method for data condensation in their adaptation of fractionation hierarchical clustering (Cutting, Karger, Pedersen, & Tukey, 1992) to hierarchical model-based fractionation and refractionation algorithm. The basic idea of model-based fractionation is to split the data set into fractions, and then compresses each fraction into a fixed number of clusters using hierarchical model-based clustering method. These resulting clusters are termed as meta-observations which are characterized by the similar prototypes as sufficient EM, i.e. triples of sufficient statistics $(\bar{x}, S, n)$. Refractionation is applied to further splitting the fractions. Then, the final meta-observations from all the fractions are clustered. Using model-based clustering to compress data may help to preserve the clusters structure better. However, splitting the meta-observations again and again may cause more and more loss of information.

Data condensation speed up large data clustering, but it has the weakness of not letting the wrongly grouped data points to migrate once condensation is performed and causing loss of information. However, by prudent consideration of the choice of data condensation method, the loss of information can be minimized. Steiner et al. (2007) suggested an acceptable loss of information is granted by compact prototypes, and condensing data according to the observed structure of the data can generate these compact prototypes. This thesis uses Gaussian-based mixture model to condense data as it has the advantage to model clusters of different shapes, volumes and orientation, furthermore most of the data can be approximated to Gaussian distribution (McLachalan & Basford, 1988; Banfield et al., 1993; Celeux & Govert, 1995; Dasgupta & Raftery, 1998), thus, the resulting prototype should be more compact than the prototypes generated by other data condensation methods.

2.2.2.3 Prototypes of Condensed Data

From a practical point of view, the types of prototype of the condensed data should be as simple as possible to reduce memory storage. For instance, a duplet prototype needs less storage than a triple prototype. However, the sufficiency of the prototype should be also taken into account. Thus, prototype that is able to represent and characterize the original structure of the condensed data, and need less memory storage is desirable. Prototype characterized by triple sufficient statistics has been widely used because it can be accommodated in most of the clustering methods. This thesis chooses to represent the prototypes of the condensed continuous data and mixed data using triple and quadruple of maximum likelihood estimates (MLEs) of the mixture models used to condense the data respectively. The resulting MLEs are used to update the model parameters through the proposed bi-mixture models, and can be used in the proposed model selection criterion to determine change in clusters structure.

2.3 Change in Clusters Structure

The development of clustering algorithms for data mining is now concentrating on the ability of incorporation of new data from sequential scans into the trained model. The main advantage of this property is that it enables the algorithms to handle data sets that are too huge to be loaded into the memory buffer to be processed as a whole. Furthermore, with the expansion of the Internet, data arrives at explosive rate continuously from open stream. It is important to incorporate new data to keep the trained model up to date so that it would never lose potentially valuable information. However, the main challenge in incorporation of new data into the trained model is to consider the probable change in clusters structure between the sequential scans.

To maintain a reliable model in clustering open data stream, Lee, Cheung, and Kao (1998) proposed a dynamic sampling technique to detect if sufficient change has occurred in the structure of a data set, then the trained model is re-estimated using the full set of available data. However, like most of the existing algorithms, the retraining of the learned model using the data set at hand together with the newly added data complicates the situation as the large data set has grown even larger, and it is a waste of resources because the information from the previous model is not being used to build the new model. Within the framework for mining open data stream proposed by Domingos and Hulten (2003) which takes the change in clusters structure into consideration, massive stream version of *k*-means clustering (Domingos & Hulten, 2001), and EM algorithm for mixture of Gaussian (Domingos & Hulten, 2002) have been designed and implemented. The basic concept of the algorithms is to train model from finite data in finite time that is essentially equivalent to the one which would be obtained from infinite data. The relative loss between the finite data and infinite data models is bounded as a function of the number of data points used at each learning step of the finite data, and the number of these data points is minimized subject to target limits on the loss. The algorithm assumes that the data points are independent and identically distributed. However, many data stream evolves over time.

Some sequential clustering algorithms have been proposed for clustering large data with the consideration of change in clusters structure. Hartigan (1975) proposed single pass sequential algorithm where a sample is first trained and the remaining data points are classified one by one in the trained model. If the distances of a data point to all existing clusters exceed a fixed threshold, the data point will be placed in a new cluster. Kaufman et al. (1990) highlighted several shortcomings of this method: 1) the number of clusters found is not certain, 2) the first few clusters are usually much larger than the later ones since they get first chance when each object is allocated, 3) the results are dependent on the input order of objects. A model-based version of the similar multistage mechanism, which updates change of clusters structure using a modified likelihood ratio test statistic, is shown in Maitra

(2001) with application in software metrics and tomography. Although it manages to incorporate the data incrementally added to update the clusters structure and recover new clusters, it also suffers from the problem of identifying too many clusters due to the assumption of common covariance, a requirement of the modified likelihood ratio test. Furthermore, clustering the remaining data one by one is considered too slow. This thesis develops a scalable and exhaustive algorithm for clustering data sets that is too huge to be all loaded in the memory buffer in a spirit similar to the multistage algorithm of Maitra (2001) but using different method to test new clusters.

2.4 Estimating the Number of Clusters for Condensed Data

In most of the clustering algorithms, the number of clusters is assumed known. However, in practice the number of clusters in a given data is always not known in advance. Assessing the number of clusters is an important but very difficult task especially when the data set is not being processed as a whole. In finite mixture model, the number of components of the mixture model corresponds to the number of clusters. A straightforward approach is to use likelihood ratio test to formulate the number of mixture components in terms of a test on the smallest number of components in the mixture model compatible with the data. However, this test statistic does not have the usual asymptotic null distribution of chi-square in mixture models (Wolfe, 1971). McLachlan (1987) proposed to use bootstrapping approach to assess the null distribution of the likelihood ratio test statistic. An alternative approach is to use an approximation to twice the log Bayes factor called Bayesian Information Criterion (BIC) to determine the number of clusters from the model that maximized the likelihood. Other approaches exist for choosing the number of components based on model likelihood. Banfield et al. (1993) proposed the approximate weight of evidence (AWE) as an alternative to BIC. However, Fraley et al. (2002) had shown that AWE performs consistently worse than BIC, and not comparable between models with different restriction level. Instead of BIC and AWE, alternative approaches have been used to estimate the integrated likelihood (Cheeseman & Stutz, 1995; Evans, Alder & DeSilva, 2003).

There is still very limited work done on developing alternative methods to estimate the number of clusters in condensed data. Most of the clustering algorithms designed for condensed data assumed the number of clusters is known. Steiner et al (2007) proposed a variant of BIC, which takes sufficient likelihood as an approximation of the likelihood of the original data. Unfortunately, even when the additional conditions for the termination for sufficient EM, i.e. the sums of absolute changes in the mixture parameters respectively must be smaller than a set of thresholds, are held, the values for the variant of BIC for a range of number of clusters, *G*, can be in an irregular trend, and the number of clusters for the condensed data is determined based on the minimum value for the given range of *G*. It chooses incorrect model if the given range of *G* does not consist of the true number of clusters. Based on the study of Wolfe (1971) that the degree of freedom of the likelihood ratio test would be approximated by twice the difference in the number of parameters, Chou and Reichl (1999) developed a penalized BIC with weight of penalty equal to 2 for model compression criterion for decision tree state tying. Their experimental results found that the penalized BIC can be used as a more effective model compressing method compared to BIC which leads to overgrown tree.

2.5 Mixed Variables Model-based Clustering

Since the inception of incorporating finite mixture model in clustering, it has received great interest from the statistics community and turns out to be one of the most successful applications. The application of finite mixture model in clustering depends on the types of data (i.e. nominal, ordinal or numerical) or the types of variables (i.e. categorical or continuous). Wolfe (1967, 1970) first suggested to using normal mixture model for clustering continuous variables. The model is described fully by McLachlan (1982), and McLachlan and Basford (1998). For binary data clustering, Lazarsfeld and Henry (1968) introduced a mixture of Bernoulli densities which is also known as the traditional latent class model. The latent class model was then extended to nominal variables (Goodman 1974a, 1974b) and ordinal data (Clogg, 1988; Heinen, 1996). However, these mixture models have the limitation of not being able to cluster mixed data (or mixed variables data), and the drawback is critical because in real life applications, clustering always confronts with mixed data. It is common to transform the mixed variable to either only categorical or continuous variable, and then applies clustering method according to the obtained type of variable. Apparently, which ever way of such transformation, loss of information occurs.

To consider both categorical and continuous variables in clustering, Everitt (1988) proposed the underlying variable mixture model in which the categorical variables are assumed to have arisen through threshold of unobservable continuous variables, and the observed continuous variables are assumed to be jointly multivariate Gaussian. In practice, the method is limited to one or two categorical variables (Everitt and Merette, 1990). For q categorical variables, it requires q-dimensional integration at each iteration of the EM algorithm, and therefore computationally intractable for large q. As an alternative, Lawrence and Krzanowski

(1996) proposed a finite mixture model for mixed-mode data which assumes the cluster conditional densities conform to the conditional Gaussian model for mixed variable. In contrast to the threshold approach considered by Everitt (1988), the conditional Gaussian model does not impose any orders of the categories in each categorical variable and any structure on the conditional means. There are few assumptions in the conditional Gaussian model: 1) the number of clusters is known, 2) the distribution of the continuous variables is conditioned by the location, and (3) the covariance matrix is the same throughout all locations and clusters. The drawback of the conditional Gaussian model is that its great flexibility leads to multiple distinct equally likely solutions of the likelihood equations (Willse & Boik, 1999). Willse et al. (1999) proposed to impose restrictions on the conditional means of the continuous variables to solve the non-identifiability problem in the conditional Gaussian model. A modified location model (MLM) is proposed by Franco, Crossa, Villasenor, Taba, and Eberhart (1998) to tackle the problems of empty cells arising in the conditional Gaussian model. The MLM assumes: 1) the mean vectors of continuous variable do not depend on the multinomial cell, but on the clusters. The MLM uses the information from the full cells to compute an estimator of the mean and variance of each cluster that are weighed by the number of observations in the cells, 2) independence between the continuous variables and the collapsed categorical variable, 3) heterogeneity or homogeneity covariance matrices across clusters.

A variant of the conditional Gaussian model had been shown in the works by Jorgenson and Hunt (1996) and Hunt and Jorgenson (1999), where a general class of mixture models to include data having mixed categorical and continuous variables was proposed. The model is a joint generalization of both latent class models and mixtures of multivariate normal distribution. By assuming local independence, they suggested to partitioning an observed vector of variables where the variables within partition cell are independent of the variables in the complementary partition cell. In the study of normalized Gaussian expert network, Ng and McLachlan (2008) adopted the conditional Gaussian model by considering some dependence between the categorical and continuous variables.

The mixture models for clustering mixed data mentioned above are confined to small data sets. To the best of the author's knowledge, there is no mixture model clustering algorithm developed for very large mixed data sets. In this thesis a mixture model for mixed data clustering is proposed, and it is used for compressing mixed data into summarized information in a quadruplet of MLE of the sufficient statistics, mixture proportion and proportions of observations from a location conditional on a cluster. The proposed model is then incorporated into a scalable framework to develop a scalable clustering algorithm for mixed data.

3

Determining the Number of Components in Bi-mixture model

This chapter extends the idea of partial classification to the problem of updating a trained mixture model on the basis of unclassified data that is drawn from or outside the underlying population. The first part of this chapter proposes a model selection criterion, modified Bayes factor (MBF), for determining change in clusters structure between two Gaussian mixture models. The second part proposes a distribution, bi-Gaussian mixture model, for the incorporation of the model fitted on the unclassified data into the trained model. Simulations are carried out to compare the results obtained using the proposed bi-Gaussian mixture model and the Gaussian mixture model fitted on the combination of the training and unclassified data. The proposed bi-mixture model will be applied in the next chapter to develop a scalable clustering algorithm which allows incorporation of incremental compressed information into the in-memory model with the flexibility of changing the clusters structure and number of clusters.

3.1 Semi Supervised Learning and Partial Classification

The challenge in classical supervised learning is to construct classification rule base on labelled data that can be then applied to classify unlabelled data accurately. One of the solutions to this problem is to incorporate unlabelled or unclassified data to improve classification accuracy. The use of unlabelled data in this case is often referred to semi-supervised learning, and there has been plenty of works showing improvement in classification accuracy, for instance, in mixture model classification (Ganesalingam & McLachlan, 1978; O'Neill, 1978; McLachlan & Ganesalingam, 1982; Nigam, McCallum, Thrun & Mitchell, 2000; and Dean, Murphy & Downey, 2004), and mixture model regression (Liang, Mukherjee & West, 2007). However, Cozman and Cohen (2002), and Cozman, Cohen and Cirelo (2003) pointed out that the contribution of unlabelled data in reducing classification error actually depends on whether the trained model is correct for the unlabelled model. If the trained model is correct, both labelled and unlabelled data contribute to reduction in classification error by reducing variance under maximum likelihood estimation. On the other hand, when the trained model is incorrect for the unlabelled data, incorporation of unlabelled data leads to an increase in classification error. To remedy the dip in performance due to the problem where the mixture components are not in precise correspondence with the class labels for the unlabelled data, Nigam et al. (2000) suggested using EM-λ to reduce the weight of the unlabelled data. However, this has defeated the purpose of incorporating unlabeled data for more accurate classification rule. In image processing, Shanshahani and Langrebe (1994) also speculated that degradation of classification accuracy by incorporating unlabelled data is due to deviations from modelling assumptions such as the existence of data points from unknown classes and outliers in the unlabelled data. They suggested that unlabelled data should only be used if the trained model produces poor classification accuracy.

In a similar flavour of semi-supervised learning, McLachlan and Basford (1988) and McLachlan (1992) considered the context of partial classification where the discriminant rule of mixture model is updated using unclassified data. The mixture model is fitted on the basic of the classified training data $\mathbf{x}_j = (\mathbf{x}_1, ..., \mathbf{x}_n)$, and the unclassified data $\mathbf{x}_j = (\mathbf{x}_{n+1}, ..., \mathbf{x}_{n+m})$ via EM algorithm. The log-likelihood of the parameters formed from both classified training data and unclassified data is given by

$$\begin{aligned}\log L(\Psi) &= \sum_{k=1}^{G}\sum_{i=1}^{n} z_{ik} \log(\pi_k f_k(\mathrm{x}_i \mid \theta_k)) + \sum_{i=n+1}^{n+m} \log f_X(\mathrm{x}_i, \Psi) \\ &= \sum_{k=1}^{G}\sum_{i=1}^{n} z_{ij} \log(\pi_k f_k(\mathrm{x}_i \mid \theta_k)) + \sum_{i=n+1}^{n+m} \log \sum_{k=1}^{G} \pi_k f_k(\mathrm{x}_i, \theta_k) \\ &= \sum_{k=1}^{G}\sum_{i=1}^{n+m} z_{ik} \log(\pi_k f_k(\mathrm{x}_i \mid \theta_k)) + \sum_{k=1}^{G}\sum_{i=n+1}^{n+m} z_{ik}\pi_k \ . \qquad (3.1)\end{aligned}$$

The update of the parameters depends on the sampling scheme for the *n* classified data. Under mixture sampling scheme, assume there are n_k observations $\mathbf{x}_{jk}$ (j = 1,…,n_k) known to come from the *k*-th cluster, the MLE of parameters satisfy

$$\hat{\pi}_i = \left(n_k + \sum_{j=1}^{m} \hat{z}_{jk} \right) \Big/ (m+n), \qquad (3.2a)$$

$$\hat{\mu}_i = \left(\sum_{j=1}^{n_k} \mathbf{x}_{jk} + \sum_{j=1}^{m} \hat{z}_{jk}\mathbf{x}_j \right) \Big/ \left(n_k + \sum_{j=1}^{m} \hat{z}_{jk} \right), \qquad (3.2b)$$

$$\hat{\Sigma}_k = \frac{\sum_{j=1}^{n_k}(\mathbf{x}_{jk} - \hat{\mu}_k)(\mathbf{x}_{jk} - \hat{\mu}_k) + \sum_{j=1}^{m} \hat{z}_{jk}(\mathbf{x}_j - \hat{\mu}_k)(\mathbf{x}_j - \hat{\mu}_k)}{(n_i + \sum_{j=1}^{m} \hat{z}_{jk})} \qquad (3.2c)$$

for $k = 1,\ldots,g$. The posterior probability that $\mathbf{x}_j$ belongs to the *k*-th cluster is given by

$$z_{jk} = \frac{\pi_k \,|\Sigma_k|^{-1/2} \exp\{-1/2(\mathbf{x}_j - \mu_k)'\Sigma_k^{-1}(\mathbf{x}_j - \mu_k)\}}{\sum_t \pi_t \,|\Sigma_t|^{-1/2} \exp\{-1/2(\mathbf{x}_j - \mu_t)'\Sigma_t^{-1}(\mathbf{x}_j - \mu_t)\}} \qquad (3.2d)$$

In the case where the classified data provides no information on the mixing proportions, the following equation for the $\hat{\pi}_i$

$$\hat{\pi}_i = \left(\sum_{j=1}^{n} \hat{z}_{jk} \right) \Big/ n, \qquad (3.3a)$$

should be used in conjunction with (3.2b – 3.2c).

The assumption in the partial classification is that the unclassified data points are with respect to at least one of the clusters from the mixture model of the classified data. However, if there is change of cluster structures between the period when sampling of classified and unclassified data are made, the unclassified data may be from a mixture of clusters that are different from the mixture model of the classified data. McLachlan (1992) pointed out that this problem needs to be addressed differently.

This thesis extends the above partial classification to the problem of updating a trained mixture model on the basis of unclassified data that has been drawn from or outside the underlying population. The unclassified data is first fitted into a separated mixture model itself. A model selection criterion is proposed to determine whether the unclassified data has the same distribution as the model trained by the classified data, and then the trained model is updated to the proposed bi-mixture model. The update of parameters in the proposed bi-mixture model involves only the MLEs of the two mixture models fitted from classified training data and unclassified data respectively. In this chapter, the classified training data and unclassified data are fitted into Gaussian mixture models, and therefore the bi-mixture model is the specific case of the bi-Gaussian mixture model.

3.2 Detecting Change in Clusters Structure: A Modified Bayes Factor

Let the classified training data of size n_c, $\mathbf{x}_j = (\mathbf{x}_1, ..., \mathbf{x}_{n_c})$, be known to come from a g_c-component Gaussian mixture models, M_c, and the unclassified data of size n_u, $\mathbf{x}_j = (\mathbf{x}_1, ..., \mathbf{x}_{n_u})$, be fitted into a g_u-component Gaussian mixture models, M_u. To incorporate M_u into M_c, the first concern is to determine whether the clusters from M_u are in respect to the clusters from M_c. Under the bi-mixture model framework, a model selection criterion is proposed to determine whether the cluster structures have changed between the time the classified training data and the unclassified data are drawn. The change of clusters structure is detected by checking the representativeness (or distinctiveness) of the clusters identified in model M_c. Each cluster from M_u is compared to all the clusters from M_c, but at each time a cluster from M_c and a cluster from M_u are paired for comparison. If a cluster form M_u is found to come from an identified cluster from M_c, it implies that the structure of the particular cluster from M_c is not changed between the time when sampling of classified and unclassified data are made, and therefore the parameters of the respective cluster from M_c are updated on the basis of the MLEs of the cluster from M_u. Otherwise, the clusters structure of M_c is considered changed, and the MLEs of the cluster from M_u are added to model M_c as a new mixture component. The proposed model selection criterion is developed based on the MLEs of the mixture components. For this purpose, both models M_c and M_u are first decomposed into the respective mixture components and each component is represented by its MLEs as shown as below.

Approximated MLEs for Mixture Components

A fitted mixture model consists of the MLE set for the model. To obtain the MLEs of each of the mixture component, an approximation is considered as follows. Let the g-component mixture model fitted from the observations data $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ has the clusters data $(\mathbf{x}_{1k},...,\mathbf{x}_{n_k k})$ of cluster sizes n_k where k = 1,… , g. The maximum complete data log-likelihood is decomposed by grouping the terms according to clusters can be rewritten as

$$\log L(\hat{\boldsymbol{\Theta}},\hat{\mathbf{z}}\,|\,\mathbf{x}) = \sum_{i=1}^{n_1} \hat{z}_{i1} \log(\hat{\pi}_1 f_1(\mathbf{x}_{i1}\,|\,\hat{\theta}_1) + \ldots + \sum_{i=1}^{n_g} \hat{z}_{ig} \log(\hat{\pi}_g f_g(\mathbf{x}_{ig}\,|\,\hat{\theta}_g)\,. \quad (3.4)$$

The assignment of observations to the mixture components is done according to equation (1.11), and equation (3.4) can be simplified as

$$\begin{aligned}\log L(\hat{\boldsymbol{\Theta}},\hat{\mathbf{z}}\,|\,\mathbf{x}) &= \sum_{i=1}^{n_1}\{\log\hat{\pi}_1 + \log f_1(\mathbf{x}_{i1}\,|\,\hat{\theta}_1)\} + \ldots + \sum_{i=1}^{n_g}\{\log\hat{\pi}_g + \log f_g(\mathbf{x}_{ig}\,|\,\hat{\theta}_g)\} \\ &= \sum_{i=1}^{n_1}\log f_1(\mathbf{x}_{i1}\,|\,\hat{\theta}_1) + \ldots + \sum_{i=1}^{n_g}\log f_g(\mathbf{x}_{ig}\,|\,\hat{\theta}_g) + \sum_{k=1}^{g} n_k \log\hat{\pi}_k \\ &= \log L(\hat{\theta}_1\,|\,\mathbf{x}_{n_1 1}) + \ldots + \log L(\hat{\theta}_g\,|\,\mathbf{x}_{n_g g}) + \sum_{k=1}^{g} n_k \log\hat{\pi}_k \qquad (3.5)\end{aligned}$$

where $\log L(\hat{\theta}_k\,|\,\mathbf{x}_{n_k k})$ is the approximated maximum log-likelihood for the *k*-th cluster.

Therefore, instead of estimating the MLEs for the *k*-th cluster, $\hat{\theta}_k$, from maximizing the log-likelihood function of the *k*-th cluster using its data points, $\hat{\theta}_k$ can be approximately estimated from the decomposition of the maximum complete log-likelihood function of the whole set of observation data. Let the MLE of the parameter set for *M_c* be $\hat{\boldsymbol{\Psi}} = \{\hat{\pi}_1,\ldots,\hat{\pi}_{g_c},\hat{\theta}_1,\ldots,\hat{\theta}_{g_c}\}$, where $\hat{\theta}_k = (\hat{\mu}_{ck},\hat{\Sigma}_{ck})$, and the MLE of the parameter set for M_u be $\hat{\boldsymbol{\Theta}} = \{\hat{\lambda}_1,\ldots,\hat{\lambda}_{g_u},\hat{\alpha}_1,\ldots,\hat{\alpha}_{g_u}\}$, where $\hat{\alpha}_k = (\hat{\mu}_{uk},\hat{\Sigma}_{uk})$. Consider $\hat{\boldsymbol{\Psi}}$ is decomposed according to equation (3.5) into $\hat{\Psi}_{ck} = (\hat{\mu}_{ck},\hat{\Sigma}_{ck},n_{ck})$, where the cluster size for cluster *k* from model M_c is given by $n_{ck} = \hat{\pi}_k n_c$, and *k* = 1, …, g_c. Similarly, $\hat{\boldsymbol{\Theta}}$ is decomposed into $\hat{\Theta}_{uk} = (\hat{\mu}_{uk},\hat{\Sigma}_{uk},n_{uk})$, where the cluster size for cluster *k* from model M_u is given by $n_{uk} = \hat{\lambda}_k n_u$, and *k* = 1, …, g_u.

Concentrated Log-likelihood Function

Given a set of observations **x** = (**x**$_1$, ..., **x**$_n$) from a multivariate normal distribution, *N*((μ, Σ), the likelihood has the form

$$L(\mu,\Sigma)=\prod_{i=1}^{n}(2\pi)^{-\frac{p}{2}}\,|\,\Sigma\,|^{-\frac{1}{2}}\,\exp\left\{\frac{1}{2}(\mathbf{x}_i-\mu)^T\Sigma^{-1}(\mathbf{x}_i-\mu)\right\}.$$

Let the maximum likelihood estimator of $\mu=\hat{\mu}$ and $\Sigma=\hat{\Sigma}$, the maximum log-likelihood is

$$\begin{aligned}\log L(\hat{\mu},\hat{\Sigma})&=-\frac{np}{2}\log(2\pi)-\frac{n}{2}\log\left|\hat{\Sigma}\right|-\frac{1}{2}\sum_{i=1}^{n}(x_i-\hat{\mu})^T\Sigma^{-1}(x_i-\hat{\mu})\\&=-\frac{np}{2}\log(2\pi)-\frac{n}{2}\log\left|\hat{\Sigma}\right|-\frac{1}{2}trace(W\hat{\Sigma}^{-1})\end{aligned}$$

where $W=\sum_{i=1}^{n}(x_i-\hat{\mu})^T(x_i-\hat{\mu})=\hat{\Sigma}$. Thus the concentrated log-likelihood is

$$\log L(\hat{\mu},\hat{\Sigma})=\frac{-np}{2}\log(2\pi)-\frac{n}{2}\log|\,\hat{\Sigma}\,|-\frac{np}{2}. \qquad (3.6)$$

Therefore, for each component or cluster represented by its MLEs, the log-likelihood function for the cluster can be simplified to its concentrated log-likelihood function as (3.6).

Modified Bayes Factor (MBF)

Each of the clusters from M_u is compared to every cluster in M_c to determine whether it should be merged to one of the existing clusters or considered as a new cluster. This notion actually implies the choice between the models with the number of clusters k = 1 and k = 2 for each cluster i from M_u, where i = 1,…, g_u, when compared to all the clusters in M_c.

M_1: k = 1, i.e. pair of clusters ~ $N(\hat{\mu}_m,\hat{\Sigma}_m)$,

M_2: k = 2, i.e. cluster i ~ $N(\hat{\mu}_{ui},\hat{\Sigma}_{ui})$, cluster j ~ $N(\hat{\mu}_{cj},\hat{\Sigma}_{cj})$, for $j = 1,\ldots,g_c$, where $(\hat{\mu}_m, \hat{\Sigma}_m)$ are the MLEs of the merged clusters (see equations (3.15a – c)).

This thesis chooses the Bayesian approach based on Bayes factor for the above pair wise models comparison as it has advantages over the alternative frequentist hypothesis testing in the general context of model comparison. The Bayes factor (Jeffreys, 1935; 1961) is a methodology for quantifying evidence in favour of one hypothesis H_1 over another H_2. The subject has been reviewed in detailed by Kass & Raftery (1995). For a data set **x**, let the prior probabilities of the hypotheses be given by $p(H_1)$ and $p(H_2)$ respectively. From Bayes theorem, the posterior probabilities for the hypotheses are given by

$$p(H_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid H_i)p(H_i)}{\sum_j^2 p(\mathbf{x} \mid H_j)p(H_j)}, \qquad \text{for } i = 1, 2, \qquad (3.7)$$

The posterior odd can be simplified to

$$\frac{p(H_1 \mid \mathbf{x})}{p(H_2 \mid \mathbf{x})} = \frac{p(\mathbf{x} \mid H_1)}{p(\mathbf{x} \mid H_2)} \frac{p(H_1)}{p(H_2)},$$

and the transformation will give

$$B_{12} = \frac{p(M_1 \mid \mathbf{x})}{p(M_2 \mid \mathbf{x})} \div \frac{p(M_1)}{p(M_2)} = \frac{p(\mathbf{x} \mid M_1)}{p(\mathbf{x} \mid M_2)} \qquad (3.8)$$

which is the Bayes factor, B_{12}. Thus, Bayes factor is given by the ratio of the posterior odds to its prior odds in favour of H_1 over H_2.

In both cases where the two hypotheses are: 1) single distribution with no free parameters, and 2) there are unknown parameters under either or both hypotheses, the Bayes factors are given by (3.8) which is the likelihood ratio. However, in the latter case, the densities $p(\mathbf{x} \mid H_i)$ for i = 1, 2, are obtained by integrating (not maximizing) over the parameter space given by

$$p(\mathbf{x} \mid H_i) = \int p(\mathbf{x} \mid \theta_i, H_i)\pi(\theta_i \mid H_i)d\theta_i, \qquad (3.9)$$

where θ_i is the parameter under H_i, $\pi(\theta_i \mid H_i)$ is the prior density of the parameter, and $p(\mathbf{x} \mid \theta_i, H_i)$ is the probability density of **x** given θ_i, or the likelihood function of θ_i.

In practice, the marginal probability of the data, also termed as marginal likelihood or integrated likelihood, obtained from (3.9) is often difficult to compute. Schwarz (1978) proposed to penalize the log-likelihood to approximate the integrated likelihood, resulting in an approximation to the log Bayes factor also known as the Schwarz criterion and is given by

$$\mathbf{SC} = L(\hat{\theta}_1 \mid \mathbf{x}) - L(\hat{\theta}_2 \mid \mathbf{x}) - \frac{1}{2}(m_1 - m_2)\log n, \qquad (3.10)$$

where $\hat{\theta}_i$ is the MLE under H_i, m_i is the dimension of θ_i, for $i = 1, 2$.

One of the variants of the Schwarz criterion is the well known Bayes Information Criterion (BIC) where BIC = 2SC. In model selection framework, the BIC is used to compare each probable model H_1 to the constantly unchanged hypothetical model H_2. Thus, the BIC reduces to (2.12).

In Bayesian applications, pair wise models comparison is always based on the Bayes factor. Smith and Spiegelhalter (1980) had extended the Bayes factor for a standard comparison of nested hypotheses in the general linear model in the p-dimensional multivariate normal case with the following approximation:

$$-2 \log \mathrm{B}_{r,\, r+1} = \lambda - \left\{\frac{3}{2} + \log[\rho(n_{r,r+1})]\right\} \delta_{r,\, r+1}, \qquad (3.11)$$

where λ is the likelihood ratio test statistic, $\delta_{r,r+1}$ is the degree of freedom in the asymptotic chi-square distribution of λ, $n_{r,r+1}$ is the number of observations in the merged cluster, and $\rho(n_{r,r+1})$ is the rate of "shrinkage" of the prior covariance matrix which can be approximated to $n_{r,r+1}$ when $n_{r,r+1}$ is large. Unfortunately, the regularity conditions do not hold for λ to have its usual asymptotic null distribution of chi-squared with the degree of freedom $\delta_{r,r+1}$ in the clustering context. Based on a small scale simulation study of multivariate normal component densities with common covariance matrix for the number of clusters k = 1 versus k = 2, Wolfe (1971) suggested an approximation of $2\delta_{r,r+1}$ to get around the problem. However, McLachlan (1987) showed that Wolfe's approximation can be misleading for heteroscedastic normal component distributions if the outcome of the test is rigidly interpreted from the too small estimated p-value. This leads to the selection of more complicated model and overestimation of the number of clusters. Using the conclusion from Everitt (1981) that Wolfe's approximation performs well for $\delta_{r,r+1}$ between the values of 1 and 5, Banfield et al. (1993) developed the approximate weight of evidence (AWE) for the estimation of the number of clusters, which avoids problem based on significance testing. However, Fraley et al. (2002) had shown that AWE performs consistently worse than BIC, and not comparable between models with different restriction level.

In this thesis, the decision on whether the cluster structure has been changed for each of the cluster pairs is related to the choice between the models with the number of clusters k = 1 and k = 2. Therefore, this thesis adopts a special case of the extended Bayes factor from equation (3.11) where r = 1 with Wolf's approximation. It further assumes that the merged cluster size is large for massive data clustering to approximate the Bayes factor as follows

$$-2 \log \mathrm{B}_{r,\,r+1} = \lambda - \left\{\frac{3}{2} + \log(n_{r,r+1})\right\} 2\delta_{r,\,r+1}. \tag{3.12}$$

For the case where the choice is between the models with the number of clusters $k = g$ and $k = g + 1$ where g>1, the term λ considers only the likelihood of the clusters involved in the merger as the likelihoods for the clusters that are not involved in the merger cancel out in the likelihood ratio. Thus, it is sufficient for the proposed model selection criterion to consider only the pair of clusters being hypothesized in the merger. Let the maximum log-likelihood for the pair of clusters be log L_i and log L_j respectively, and the maximum log likelihood for the cluster resulting from the merger of the pair of clusters be log L_m. Therefore, the term λ can be written as

$$\lambda = 2(\log L_i + \log L_j - \log L_m), \tag{3.13}$$

where log L_i, log L_j and log L_m can be obtained by substituting the respective MLEs in equation (3.6). Substituting equation (3.13) in (3.12), the extended Bayes factor will now become the proposed modified Bayes factor (MBF) given by

$$-2 \log \mathrm{B}_{r,\, r+1}$$
$$= -n_{1i} \log\left|\hat{\Sigma}_{1i}\right| - n_{2j} \log\left|\hat{\Sigma}_{2j}\right| + n_m \log\left|\hat{\Sigma}_m\right| - 2\left(d + \frac{d(d+1)}{2}\right)\left\{\frac{3}{2} + \log(n_m)\right\} \tag{3.14}$$

The MBF suggests the choice of models based on the change in log-likelihood as a result of merging the pair of clusters. From (3.6), it can be seen that the smaller the generalize variance $|\hat{\Sigma}|$ the larger is the log-likelihood. Therefore, for each cluster from M_u, if the MBFs are positive when paired with all the clusters from M_c, the merged clusters give bigger generalize variances and smaller log-likelihoods (more negative) than the pairs of clusters. This suggests that all the pairs of clusters should not be merged. In other words, the clusters structure in M_c has been changed, and the cluster from M_u is a new cluster to be added to the trained model. Otherwise, the cluster from M_u should be merged with the pair of clusters from M_c that give negative value of MBF.

3.3 Properties of MBF

The proposed model selection criterion of MBF has the following properties:

- It is used for pair of clusters
- It does not need a specific threshold for merging clusters. As cluster analysis is an exploratory data analysis, the information required to choose an appropriate threshold is often not available, resulting in arbitrary assumptions about similarity
- It merges a pair of clusters only if the merged cluster produces higher maximum log-likelihood. This is different from the traditional agglomerative method where the nearest neighbour clusters are automatically merged merely because they are relatively nearer compared to other clusters
- It can be used to detect change of clusters structure over time
- It uses MLEs of a pair of clusters to determine whether they are identical. Therefore, it is suitable to be used for condensed data that is characterised by the MLEs of the dense region

The ability of MBF to detect change in clusters structure between two samples is useful for clustering data sets that require sequential scans. Furthermore, it can be used as an alternative criterion for determining the number of clusters in the incrementally condensed data. The application of the proposed model selection criterion MBF will be discussed further in the next chapter.

3.4 Bi-Gaussian Mixture Model

Bringing all the defined notions together, the bi-Gaussian mixture model can be derived as follows.

Let MBF suggests that in M_u the k-th cluster of size n_{uk} where $k = 1, \ldots g_r$, ($g_r \leq g_c$) are clusters that come from the clusters already identified in model M_c, whereas the k-th cluster of size n_{uk} where $k = g_{r+1}, \ldots, g_u$, are new clusters that have not been identified so far in model M_c. The updates of MLE of the parameters for the trained Gaussian mixture model, M_c, on the basis of the clusters from M_u into a bi-Gaussian mixture model depend on the decision from MBF.

Firstly, consider the case when the MBF suggests to merging the cluster from M_u with a cluster in M_c. In the framework of bi-Gaussian mixture model, the MLEs of the merged k-th cluster $(\hat{\omega}_k, \hat{\tau}_k) = (\hat{\mu}_{mk}, \hat{\Sigma}_{mk}, \hat{\tau}_k)$ are estimated approximately through one-step sufficient EM (Steiner & Hudec, 2007) using MLEs of the pair of clusters involved in merging, $(\hat{\theta}_k, \hat{\pi}_k) = (\hat{\mu}_{ck}, \hat{\Sigma}_{ck}, \hat{\pi}_k)$ and $(\hat{\alpha}_k, \hat{\lambda}_k) = (\hat{\mu}_{uk}, \hat{\Sigma}_{uk}, \hat{\lambda}_k)$, as follows

$$\hat{\tau}_k = \frac{\hat{\pi}_k n_c + \hat{\lambda}_k n_u}{n_c + n_u}, \tag{3.15a}$$

$$\hat{\mu}_{mk} = \frac{\hat{\pi}_k n_c \hat{\mu}_{ck} + \hat{\lambda}_k n_u \mu_{uk}}{\hat{\pi}_k n_c + \hat{\lambda}_k n_u}, \tag{3.15b}$$

$$\begin{aligned} \hat{\Sigma}_{mk} = \Big(&\{\hat{\pi}_k n_c (\hat{\mu}_{ck} - \hat{\mu}_{mk})(\hat{\mu}_{ck} - \hat{\mu}_{mk}) + \hat{\lambda}_k n_u (\hat{\mu}_{uk} - \hat{\mu}_{mk})(\hat{\mu}_{uk} - \hat{\mu}_{mk})\} \\ &+ \{\hat{\pi}_k n_c \hat{\Sigma}_{ck} + \hat{\lambda}_k n_u \hat{\Sigma}_{uk}\}\Big) \Big/ \Big(\hat{\pi}_k n_c + \hat{\lambda}_k n_u\Big) \end{aligned} \tag{3.15c}$$

The complete maximum log-likelihood function of model M_c is updated by the pairs of merged clusters from M_u and it is given by

$$\log L(\hat{\Psi}) = \sum_{k=1}^{g_c} \sum_{i=1}^{n_c} \hat{z}_{ij} \log\{\hat{\pi}_k \phi_k(\mathbf{x}_i \mid \hat{\theta}_k)\} + \sum_{k=1}^{g_r} \sum_{i=1}^{n_w} \hat{z}_{ik} \log\{\hat{\lambda}_k \phi_k(\mathbf{x}_i, \hat{\alpha}_k)\} \tag{3.16}$$

where $n_w = \sum_{k=1}^{r} n_k$ is the total number of data points in all the clusters from M_u that come from the already identified clusters. (3.16) can be simplified to

$$\log L(\hat{\Psi}) = \sum_{i=1}^{n_{c1}} \log\{\hat{\pi}_1 \phi_1(\mathbf{x}_i \mid \hat{\theta}_1)\} + \ldots + \sum_{i=1}^{n_{cc}} \log\{\hat{\pi}_{g_c} \phi_{g_c}(\mathbf{x}_i, \hat{\theta}_{g_c})\}$$

$$+\sum_{i=1}^{n_{u1}}\log\{\hat{\lambda}_1\phi_1(\mathbf{x}_i \mid \hat{\alpha}_1)\}+\ldots+\sum_{i=1}^{n_{ur}}\log\{\hat{\lambda}_{g_r}\phi_{g_r}(\mathbf{x}_i,\hat{\alpha}_{g_r})\}$$

$$=\sum_{i=1}^{n_c+n_{ur}}\sum_{k=1}^{g_c}\log\{\hat{\tau}_k\phi_k(\mathbf{x}_i \mid \hat{\omega}_k)\} \qquad (3.17)$$

If the cluster from the model M_c is not being selected to merge with any cluster from M_u, the MLE of sufficient statistics of the cluster remain but the mixture proportion $\hat{\pi}_k$ is updated to

$$\hat{\pi}_k^{\,*}=\frac{\hat{\pi}_k n_c}{n_c+n_u}, \qquad (3.18)$$

and the MLEs of the cluster become $(\hat{\omega}_k,\hat{\tau}_k)=(\hat{\theta}_k,\hat{\pi}_k^{\,*})$.

Secondly, for the clusters from M_u that have not been identified in M_c, the log-likelihood is given by the linear combination of each component

$$\log L(\hat{\Lambda})=\sum_{i=1}^{n_{ug_r}}\log\hat{\lambda}_{g_{r+1}}\phi(\mathbf{x}_i,\hat{\alpha}_{g_{r+1}})+\ldots+\sum_{i=1}^{n_{ug_u}}\log\hat{\lambda}_{g_u}\phi(\mathbf{x}_i,\hat{\alpha}_{g_u})$$

$$=\sum_{i=1}^{n_u-n_w}\sum_{k=g_{r+1}}^{g_u}\log\hat{\lambda}_k\phi(\mathbf{x}_i,\hat{\alpha}_k) \qquad (3.19)$$

Combining the maximum log-likelihood functions (3.17) and (3.19) has to adjust the mixture proportion of the updated existing clusters, which is given by

$$\log L(\hat{\Omega})=\sum_{i=1}^{n_c+n_{ur}}\sum_{k=1}^{g_c}\log\{\left[1-\sum_{k=g_{r+1}}^{g_u}\hat{\upsilon}_k\right]\hat{\tau}_k\phi_k(\mathbf{x}_i \mid \hat{\omega}_k)\}+\sum_{i=1}^{n_u-n_w}\sum_{k=g_{r+1}}^{g_u}\log\hat{\lambda}_k\phi(\mathbf{x}_i,\hat{\alpha}_k) \qquad (3.20)$$

where $\hat{\upsilon}_k=\dfrac{\hat{\lambda}_k n_u}{n_c+n_u}$.

Finally, the approximated density function for the bi-Gaussian mixture model that takes account of the change in clusters structure between the classified and unclassified data will become

$$g(\mathbf{x}_i,\Omega)=\left[1-\sum_{k=g_{r+1}}^{g_u}\upsilon_k\right]\sum_{k=1}^{g_c}\tau_k\phi_k(\mathbf{x}_i,\omega_k)+\sum_{k=g_{r+1}}^{g_u}\upsilon_k\phi(\mathbf{x}_i,\alpha_k)\,. \qquad (3.21)$$

3.5 Experimental Result

In this section, two sets of synthetic data are presented to assess the effectiveness of updating trained model using unclassified data based on the proposed bi-Gaussian mixture model. The criterion used to assess the accuracy of the estimated model parameters is the classification accuracy. Three experiments were designed for this purpose. First, the generated data were divided into classified and unclassified data according to different ratios using separate sampling scheme so that the mixture proportions in both sets of data are not known. In this case, the parameters of the true model are the population parameters. Second, the generated data were used as classified data, and the unclassified data were generated from the same population parameters as the classified data except that the mixture proportions and data sizes were varying. The effect of the size of unclassified data was also studied. Lastly, the experiment was designed to study how well the proposed model selection criterion manages to identify the change in clusters structure and how accurate the model can update new cluster through bi-Gaussian mixture. For this case, new cluster that is not identical to any clusters in the classified data was added to the unclassified data, and the mixture proportions of the unclassified data were different from the classified data.

The results are compared to the model obtained using the combination of classified training data and unclassified data fitted by the Gaussian mixture model.

3.5.1 Simulation Study 1

A data set consists of 1000 data points was generated from the mixtures of four-component bivariate normal distribution considered by Figueiredo and Jain (2002). The data by Figueiredo (2002) is considered here because it is challenging in fitting the Gaussian mixture model for clusters in the mixtures which are overlapping and share a common mean but different covariance matrices. The parameters are given as follows. Cluster 1 and 2 are overlapping, and two of the four components share a common mean but different covariance matrices.

$$\pi_1 = \pi_2 = \pi_3 = 0.3\,,\ \pi_4 = 0.1\,;\ \mu_1 = \mu_2 = (-4,-4)\,,\ \mu_3 = (2,2)\,,\ \mu_4 = (-1,-6)\,, \text{ and}$$

$$\Sigma_1 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 6 & -2 \\ -2 & 6 \end{pmatrix},\ \Sigma_3 = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix},\ \Sigma_4 = \begin{pmatrix} 0.125 & 0 \\ 0 & 0.125 \end{pmatrix}.$$

Table 3.1. Classification accuracy and log-likelihood for bi-Gaussian mixture model and Gaussian mixture model in simulation study 1. The bi-Gaussian mixture model fitted on data that divided into classified and unclassified data according to different ratio. The Gaussian mixture model fitted on the combined data.

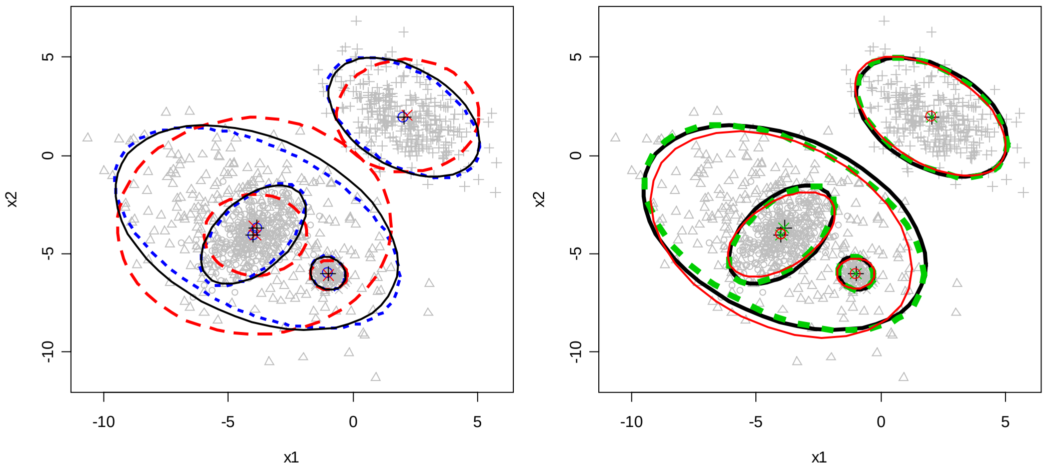
Ratio	Trained model updated by unclassified data			Model from combined data		
$n_c : n_u$	k	Classification accuracy (%)	Log-likelihood	k	Classification accuracy (%)	Log-likelihood
1:1	4	87.30	-4624.44	4	88.10	-4612.26
4:1	4	87.40	-4619.89	4	88.10	-4612.26

Table 3.2. Results for models obtained from bi-Gaussian mixture model and Gaussian mixture model in simulation study 1. Unclassified data in the bi-Gaussian mixture model are generated from different mixture proportions and with new cluster added. The Gaussian mixture model fitted on the combined data.

Unclassified data		Trained model updated by unclassified data			Model from combined data		
Mixture proportions (λ_1, λ_2, λ_3, λ_4)	n_u	k	Classification accuracy (%)	Log-likelihood	k	Classification accuracy (%)	Log-likelihood
(0.3,0.3,0.3,0.1)	1000	4	86.10	-9484.86	4	86.50	-9500.32
(0.3,0.3,0.3,0.1)	200	4	87.25	-5645.062	4	87.75	-5659.33
(0.25,0.25,0.25,0.25)	1000	4	88.30	-8921.83	4	88.55	-8915.41
(0.25,0.25,0.25,0.25)	200	4	87.25	-5514.49	4	87.92	-5540.12
(0.4,0.2,0.25,0.15)	1000	4	87.17	-5560.65	4	87.91	-5611.12
(0.4,0.2,0.25,0.15)	200	4	87.75	-9144.05	4	88.15	-9109.58
(0.25,0.25,0.25,0.10) + new cluster	200	5	86.53	-7406.80	5	87.13	-7419.90

The results for classification accuracy and log-likelihood for models obtained using the bi-Gaussian mixture model, and the Gaussian mixture model fitted on the combined data are shown in Table 3.1 and Table 3.2. It can be seen that the classification accuracies of the trained models updated by unclassified data using bi-Gaussian mixture model are virtually similar to the Gaussian mixture models fitted on the combination of the two sets of data. The log-likelihood obtained from the bi-Gaussian mixture model is very close to the Gaussian mixture model fitted on the combined data. In Table 3.1, even though the classified data and unclassified data are sampled using separate sampling scheme where there is no prior information about the mixture proportions, the bi-Gaussian mixture model successfully estimated the parameters regardless of the ratios of classified data to unclassified data. Figure 3.2 shows one of the examples where n_c: n_u = 4:1. Although the MLE of the covariances in the model fitted by the unclassified data deviate from the ones obtained by classified data, the MLE of covariances in the final model obtained using the bi-Gaussian mixture model is close to the true model.

From Table 3.2, it can be seen that the mixture proportions and size of the unclassified data do not play a significant role in the update of trained model through the bi-Gaussian mixture model. When a new cluster with parameters: μ_{u5} = (2,2), Σ_{u5} = (0.5,0.25,0.25,0.5), and λ_5 = 0.15 was added to the unclassified data, the proposed model selection criterion managed to detect it. The recovery and incorporation of the new cluster to the trained model through the bi-Gaussian mixture model gives very close results in terms of classification accuracy and log-likelihood as the model obtained using the combined data fitted by the Gaussian mixture model. Figure 3.3(a) shows the new cluster is recovered in the model fitted using the unclassified data, and Figure 3.3(b) shows the MLEs of model parameters estimated by the bi-Gaussian mixture model and the Gaussian mixture model fitted on the combined data to be very close.



(a) (b)

Figure 3.1. True clusters structure (scatter plot) and the MLE of means and covariances for the model fitted by: a) classified data ('o', blue dotted line), unclassified data ('x', red dashed line), and bi-Gaussian mixture model ('+', black solid line,), and b) bi-Gaussian mixture model ('+', black solid line), combined data ('x', green dashed line), and true model ('o', red solid line). (Note: ellipsoids visualized by 90% normal tolerance).

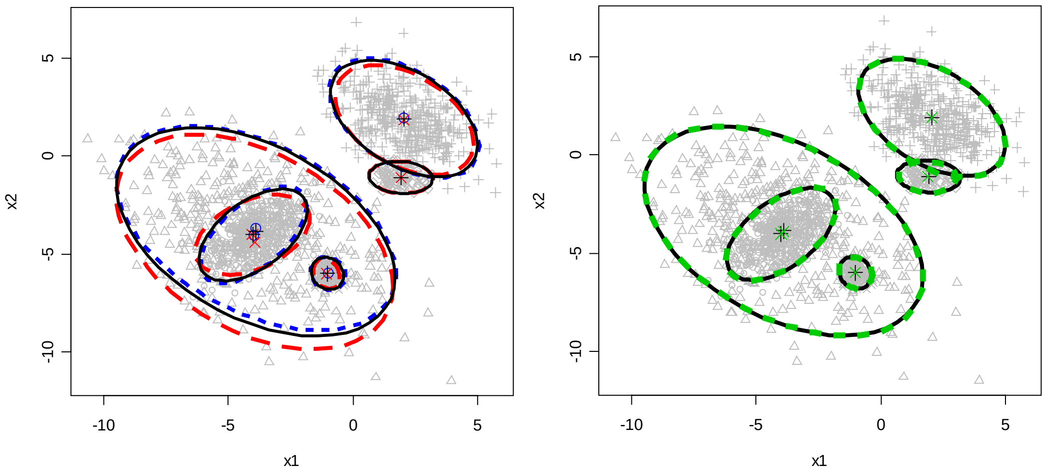


Figure 3.2. True clusters structure (scatter plot) and the MLE of means and covariances for the model fitted by: a) classified data ('o', blue dotted line), unclassified data ('x', red dashed line), and bi-Gaussian mixture model ('+', black solid line), and b) bi-Gaussian mixture model ('+', black solid line), and combined data ('x', green dashed line). (Note: ellipsoids visualized by 90% normal tolerance).

3.5.2 Simulation Study 2

A data of size 1500 was generated from a seven-component five-dimensional normal mixture. In the mixture distribution, component one and two, and also component three and four, share common mean respectively but have different covariances. The parameters are as follows.

$\pi_1 = \pi_2 = \pi_4 = \pi_6 = 0.15$, $\pi_3 = 0.175$, $\pi_5 = 0.125$, $\pi_7 = 0.1$,

$\mu_1 = \mu_2 = (0,0,0,0,0)$, $\mu_3 = \mu_4 = (4,4,4,4,4)$, $\mu_5 = (4,4,4,4,-4)$, $\mu_6 = (4,-4,-4,4,4)$,

$\mu_7 = (-4,4,4,-4,4)$, and $\Sigma_1 = \Sigma_3 = \Sigma_5 = \Sigma_6 = \Sigma_7 = I_5$, $\Sigma_2 = \Sigma_4 = 4I_5$.

Table 3.3. Table. Classification accuracy and log-likelihood for bi-Gaussian mixture model and Gaussian mixture model in simulation study 2. The bi-Gaussian mixture model fitted on data that divided into classified and unclassified data according to different ratio. The Gaussian mixture model fitted on the combined data.

Ratio	Trained model updated by unclassified data			Model from combined data		
$n_u : n_u$	k	Classification accuracy (%)	Log-likelihood	k	Classification accuracy (%)	Log-likelihood
1:1	7	90.10	-30369.32	7	89.77	-30370.85
3:1	7	89.30	-30275.29	7	89.77	-30370.85

Table 3.4. Results for models obtained from bi-Gaussian mixture model and Gaussian mixture model in simulation study 2. Unclassified data in the bi-Gaussian mixture model are generated from different mixture proportions and with new cluster added. The Gaussian mixture model fitted on the combined data.

Unclassified data		Trained model updated by unclassified data			Model from combined data		
Mixture proportions (λ_1, λ_2, λ_3, λ_4, λ_5, λ_6, λ_7)	n_u	k	c.a* (%)	Log-likelihood	k	c.a* (%)	Log-likelihood
(0.15,0.15,0.175,0.15,0.125,0.15,0.1)	1500	7	89.83	-30286.85	7	90.04	-30279.63
(0.15,0.15,0.175,0.15,0.125,0.15,0.1)	500	7	89.49	-20116.31	7	90.01	20168.51
(1/7,1/7,1/7,1/7,1/7,1/7,1/7)	1500	7	90.59	-30134.17	7	90.72	-3012837
(1/7,1/7,1/7,1/7,1/7,1/7,1/7)	500	7	89.43	-20063.91	7	89.88	-20101.08
(0.2,0.2,0.1,0.2,0.1,0.1,0.1)	1500	7	89.03	-30837.71	7	89.17	-30829.22
(0.2,0.2,0.1,0.2,0.1,0.1,0.1)	500	7	89.74	-19195.84	7	89.79	-20331.74
(0.15,0.15,0.175,0.15,0.125,0.15,0) + new cluster	500	8	88.29	-20580.91	8	88.96	-20617.69

*Note: c.a = classification accuracy

The results for classification accuracy and log-likelihood for model obtained through the bi-Gaussian mixture model and the Gaussian mixture model on the simulated data are shown in Table 3.3 and Table 3.4. Basically, the results support the finding in simulation 1. An interesting result is shown in Table 3.3 where n_c: n_u = 1:1. It shows that the classification accuracy for the model obtained by the bi-Gaussian mixture model is slightly higher than the Gaussian mixture model fitted on the combined data. This implies that the classification accuracy for the model obtained by the bi-Gaussian mixture model is not definitely lower than the Gaussian mixture model fitted on the combined data.

The proposed model selection criterion also works well in the higher dimensional data. The model selection criterion correctly suggested that there is change in clusters structure when a new cluster generated by the parameters: π_8 = (2,2,2,2,2), Σ_8 = I, λ_8 = 0.1, was added to the unclassified data. Furthermore, the classification accuracy of the bi-Gaussian mixture model, which considered change in clusters structure, is very close to the Gaussian mixture model fitted on the combined data.

3.6 Conclusion

With the development of the proposed model selection criterion MBF in this thesis, the unclassified data can be determined whether it comes from the same distribution as the classified data, before it is being incorporated to update the trained model. The main advantage of using the framework of bi-Gaussian mixture model to update the trained model is that it allows the addition of new clusters. With this breakthrough, clustering data sets that do not fit in the memory or from open data stream can be done incrementally without missing out any important clusters. The framework is applied to develop a scalable clustering algorithm for incrementally compressed data and will be discussed in the next chapter.

4

Scalable Clustering Algorithm for Very Large Data

This chapter applies the bi-Gaussian mixture model developed in the previous chapter with some modifications in model update to propose a new scalable clustering algorithm for very large data sets that do not fit into the computer memory buffer. The clustering algorithm is known as incremental compression into flexible number of clusters (FlexClust). It is evaluated using simulated and real data. The results are compared to a few existing clustering algorithms.

4.1 Incremental Compression Into Flexible Number of Clusters (FlexClust)

Mixture model clustering and its extensions are usually confined to data sets that can be processed as a whole. However, this is not practical when the data size is larger than the memory buffer of the computer.

This thesis proposes an algorithm known as FlexClust which compresses data incrementally according to the available memory buffer using the Gaussian mixture model and incorporates the compressed information into the current model with the ability to detect small clusters. The proposed clustering algorithm can accommodate a data set of any size, and it has the flexibility of allowing changes in the clusters structure and the number of clusters by means of a modified Bayes factor (MBF).

As the critical issue in data compression is to avoid loss of information, FlexClust employs mixture modelling for data compression which has the advantages of describing clusters with a variety of shapes, detecting overlapping clusters, and automatically determining the number of clusters that best fit the compressed data. Furthermore, the summarized information in the form of maximum likelihood estimates (MLEs) of the mixture model can then be used in a proposed model selection criterion to determine the representativeness of the existing clusters and to detect changes in clusters structure due to the incrementally added data. Finally, to enhance the possibility of the inclusion of small clusters, the ideas from Maitra (2001) and Fayyad et al. (1996) is adapted to propose a scan through and select procedure. In Maitra (2001), the weight of the small clusters is increased progressively in an incremental sampling scheme to avoid missing out on the small clusters at each stage. Fayyad et al. (1996) introduced the iterated sampling where data points do not fit well are accumulated in another sample for further investigation.

4.2 The FlexClust Clustering Algorithm

The idea of the proposed algorithm is to iterate over random samples of the database and incorporate information computed from the current sample with information computed over previous samples while operating within a limited memory buffer.

The algorithm of FlexClust is summarized as follow:

1 Select data points from relatively small clusters using a scan through and select procedure: i) draw a random sample from the whole data set, D, that is stored in the hard disk space of the computer, and cluster it using k-means. Any resulting clusters with the proportions less than a threshold ε (say 0.01) are considered small clusters, ii) let the set of data points from these small clusters be Q, iii) repeat steps (i) and (ii) for a few times until there is no probable new small clusters found in step (i).

2 Replicate set Q for q times and then add a random sample from the data set which size can be fitted into the computer buffer memory. Let the data points in the buffer memory be the initial sample S_1.

3 Compression of S_1: Fit S_1 with a Gaussian mixture model, and let this be denoted by $M^{(0)}$ which will be retained in the memory and S_1 will be purged. $M^{(0)}$ is the initial model under consideration.

4 Select a random sample, S_2, where $S_2 \in D \backslash S_1$ and independent of S_1. Repeat Step 3 to get model M_I.

5 For a given cluster from model M_I, find the nearest cluster in model $M^{(0)}$ (see Section 4.3 for details).

6 Determine whether to merge the clusters or to add new cluster to the current model $M^{(0)}$ using the modified Bayes factor criterion. Update the current model $M^{(0)}$ to $M^{(1)}$ (see Section 4.3 for details).

7 If the decision is to merge the nearest neighbour pair, refinement is carried out. Find the cluster from $M^{(1)}$ that is closest to the merged cluster, and apply the MBF criterion to determine whether the merged cluster should be further merged with the existing closest cluster. Update the current model $M^{(1)}$ to $M^{(2)}$ (see Section 4.3 for details). Otherwise, go to the next step.

8 Repeat Steps 5 – 7 for all the clusters from M_I, and finish the updates for model under consideration using the compressed information from a new sample to obtain the model at iteration t.

9 Let the model at iteration t obtained from Step 8 be the model under consideration. Repeat Steps 4 – 8 in the coming iterations to include remaining samples in the storage.

The FlexClust algorithm is summarized in Figure 4.1. The basic insight is to identify clusters of data points which can be effectively summarized by their MLEs. Instead of revisiting these records, updates of model are performed over their MLEs. After each buffer refill and compression, the mixture model parameters at that point of time are updated over the MLEs of the new sample and the MLEs of the in-memory mixture model parameters. Clearly with the incremental sampling and flexible number of clusters framework, any additional data from the database can be easily accommodated to update the current model and provide the most up-to-date usable model at any point of time.

The scan through and select procedure in Step 1 partitions the data to detect any probable very small clusters or clusters which are small relative to other clusters. Step 1 is replicated q times to increase the chance to detect small clusters. For image data sets, the scan through and select procedure can be carried out by drawing samples in blocks to increase the changes of detecting any probable small clusters.

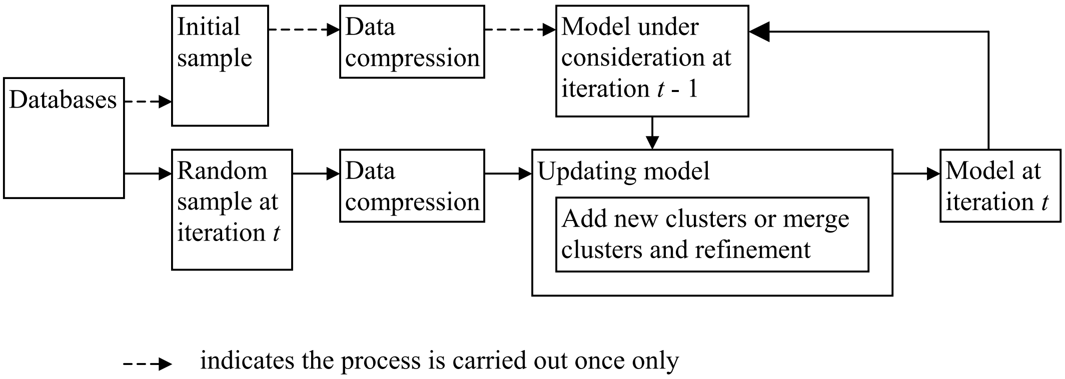

Initial sample
Data compression
Model under consideration at iteration t - 1
Databases
Random sample at iteration t
Data compression
Updating model
Add new clusters or merge clusters and refinement
Model at iteration t
indicates the process is carried out once only

Figure 4.1. The overview of FlexClust.

4.3 Components of the FlexClust Architecture

4.3.1 Incremental Sampling and Compression

In FlexClust data compression is achieved by replacing observation points at dense regions by their MLEs of the parameters of the fitted Gaussian mixture model.

Suppose S_1 = $\{ \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{n_1} \}$ is a d-dimensional initial sample of size n_1, which consists of a random sample drawn from a massive data set D together with the replicate set of probable small clusters data points detected from the scan through and select method, fills the memory buffer, is fitted into a g_1-component Gaussian mixture model using the complete log-likelihood function in (1.7). The observation points from the initial sample are compressed to a prototype system consisting of the MLEs of the parameter set, and then purged from the buffer to free some memory spaces, retaining only the prototype system. The initial model under consideration is the g_1-component Gaussian mixture model. Select an independent random sample, S_2, of size n_2, where $S_2 \subset D \setminus S_1$. Repeat the steps of fitting Gaussian mixture model, data compression, and purging of observation points. The memory buffer at this point of time contains a prototype system for the model under consideration and also a prototype system from the new sample S_2.

Let the MLEs of the parameter set for the t-th sample be $\hat{\Psi}_t = (\hat{\theta}_{t1}, \ldots, \hat{\theta}_{g_t}, \hat{\pi}_{t1}, \ldots, \hat{\pi}_{tg_t})$, where $\hat{\theta}_{tk} = (\hat{\mu}_{tk}, \hat{\Sigma}_{tk})$ is the vector consists of the MLE of mean, $\hat{\mu}_{tk}$, and full covariance matrix, $\hat{\Sigma}_{tk}$, and $\hat{\pi}_{tk}$ is the MLE of mixture proportion, for the k-th cluster from the t-th sample, where $t = 1,2$, $k = 1, \ldots, g_t$. The MLEs of each individual cluster are estimated approximately from the decomposition of the mixture model components. Thus, for the t-th sample, $\hat{\boldsymbol{\Psi}}_t$ is decomposed into its mixture components $\hat{\Psi}_{tk} = (\hat{\mu}_{tk}, \hat{\Sigma}_{tk}, n_{tk})$ for $t = 1,2$, $k = 1, \ldots, g_t$, where $n_{tk} = \hat{\pi}_{tk} n_t$ is the k-th cluster size.

4.3.2 Model Update

Nearest Neighbour Pairs

In order to minimize computational efforts, and based on the rationale that the nearest neighbour clusters are most probably identical, the model updates start with finding the nearest neighbour pairs of clusters. However, if the examination of the MLE of means within the same compression finds that there are clusters which probably share common means, multiple nearest neighbours have to be considered instead of single nearest neighbour. For each of the clusters from sample S_2, its nearest neighbour cluster from sample S_1 is determined by the shortest Euclidean distance between the MLE of means of the two clusters given by

$$d_m = \min \sum_{p=1}^{d} \left| \hat{\mu}_{1k}^{(p)} - \hat{\mu}_{2k}^{(p)} \right| , \tag{4.1}$$

for $1k$, $k = 1, \ldots, g_1$, and for $2k$, $k = 1, \ldots, g_2$.

Add Cluster or Merge Clusters and Refinement

The challenge of incremental compression into flexible number of clusters can be viewed as a problem of changing the number of clusters of the current model due to the addition of new clusters from the incremental compressed data. The proposed modified Bayes factor (MBF) from section 3.2 is used as a model selection criterion to choose the appropriate number of clusters from the combination of two sets of compressed data.

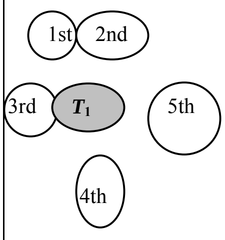
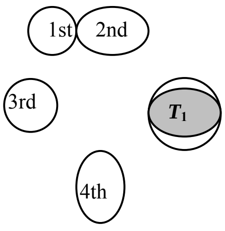
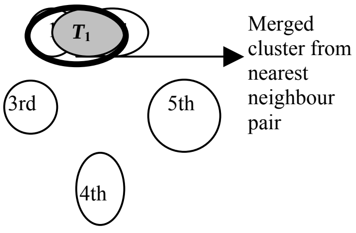
	(a) Possibility 1	(b) Possibility 2	(c) Possibility 3
Illustration			 T_1 Merged cluster from nearest neighbour pair 3rd 5th 4th
Sign of MBF	positive	negative	negative, and negative in refinement
Description	T_1 is a new cluster.	T_1 is identical to an existing cluster.	T_1 merges the 1st and 2nd existing clusters and reduces the existing number of clusters by 1.
Number of clusters	$k = G + 1$	$k = G$	$k = G - 1$

Figure 4.2. Three possibilities when a cluster T_1 is compared to the existing 5 clusters: a) add a new cluster, b) merge with an existing cluster, and c) merge two existing clusters.

When a cluster, from the later compressed data, is added to the current model with G clusters, the number of clusters is assumed to change as follows: (1) G+1, (2) G, and (3) G-1. For illustration purpose, consider a current model with G=5 as shown in Figure 4.2. Let cluster T_1 be a cluster from the later compressed data. In Figure 4.2(a), T_1 is a new cluster to be added to the current model. On the other hand, in Figure 4.2(b), T_1 is a cluster identical to an existing cluster, and the two clusters can be merged. In case Figure 4.2(c), T_1 first merges with one of the existing clusters, and then the resulting merged cluster merges with another existing cluster. The closest existing cluster to the merged cluster is determined based on the shortest Euclidean distance between the MLE of means of the two clusters.

Parameters Update

The model updates are carried out incrementally on the prototypes of the nearest neighbour pairs based on the proposed MBF model selection criterion. If MBF suggests to merge the nearest neighbour pair, the parameters of the mixture model under consideration will be updated over the cluster merged to the nearest existing cluster. Let $(\hat{\mu}_1, \hat{\Sigma}_1, n_{1a}) \in \hat{\Psi}_{1k} = (\hat{\mu}_{1k}, \hat{\Sigma}_{1k}, n_{1k})$, k = 1,…,g_1, and $(\hat{\mu}_2, \hat{\Sigma}_2, n_{2a}) \in \hat{\Psi}_{2k} = (\hat{\mu}_{2k}, \hat{\Sigma}_{2k}, n_{2k})$, k = 1,…,g_2, be the decomposed prototypes of the nearest neighbour pair, and $(\hat{\mu}_m, \hat{\Sigma}_m, n_{ma})$ be the MLE of sufficient statistics and the cluster size of the merged cluster. The parameters of the existing model are updated using weighted MLEs as follow:

1 The compressed prototype for the merged cluster is estimated from

$$n_{ma} = n_{1a} + n_{2a}, \tag{4.2a}$$

$$\hat{\mu}_m = \frac{\sum_{j=1}^{2} n_{ja}\hat{\mu}_j}{n_{ma}}, \tag{4.2b}$$

$$\hat{\Sigma}_m = \frac{\sum_{j=1}^{2} n_{ja}(\hat{\mu}_j - \hat{\mu}_m)(\hat{\mu}_j - \hat{\mu}_m) + \sum_{j=1}^{2} n_{ja}\hat{\Sigma}_j}{n_{ma}}. \tag{4.3c}$$

2 The mixture proportions of the existing model become

$$\hat{\pi}_m^{\ *} = \frac{n_{1a} + n_{2a}}{n_1 + n_{2a}}, \text{ for the component involved in merging;} \tag{4.4d}$$

$$\pi_k^{\ *} = \frac{n_{1k}}{n_1 + n_{2a}}, \quad \text{for the other components.} \tag{4.5e}$$

The model under consideration is now updated to

$$f(\mathbf{x}_i \mid \mathbf{\Psi}) = \sum_{k=1}^{g_1-1} \pi_k^{\ *}\phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k) + \pi_m^{\ *}\phi(\mathbf{x}_i \mid \mu_m, \Sigma_m) \tag{4.6}$$

$$= \sum_{k=1}^{g_1} \pi_k^{\ *}\phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k), \qquad i = 1, 2, \ldots, n_1 + n_{2a}.$$

On the other hand, if the MBF suggests a new cluster, the mixture proportions of the current mixture model will be updated and a new mixture component is added based on the assumption that the new cluster is independent of the existing clusters. The mixture proportions of the model are updated as follow:

$$\pi_{2a}^{\ *} = \frac{n_{2a}}{n_1 + n_{2a}}, \qquad \text{for the newly added cluster;} \tag{4.7a}$$

$$\pi_k^{\ *} = \frac{n_{1k}}{n_1 + n_{2a}}, \qquad \text{for the other existing components.} \tag{4.7b}$$

The model under consideration is now given by

$$f(\mathbf{x}_i \mid \mathbf{\Psi}) = \sum_{k=1}^{g_1} \pi_k^{\ *}\phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k) + \pi_{2a}^{\ *}\phi(\mathbf{x}_i \mid \mu_2, \Sigma_2) \tag{4.8}$$

$$= \sum_{k=1}^{g_1+1} \pi_k^{\ *}\phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k), \qquad i = 1, 2, \ldots, n_1 + n_{2\text{a}}$$

where $\pi_{(g_1+1)}^{*} = \pi_{2a}^{\ *}$.

4.3.3 Model at a Certain Iteration

The model at iteration *t* is built on the current model at iteration *t*-1 updated by a random sample drawn at iteration *t*. In each of the iteration for model update, the MLEs of the Gaussian mixture model under consideration are updated using the MLEs of the Gaussian mixture model fitted by the random sample drawn at iteration *t*. The model updates are performed incrementally over each nearest neighbour pair of clusters as discussed in section 4.3.2. When new random sample is being filled in the buffer memory at iteration *t*+1, the model at iteration *t* will be treated as the model under consideration that is to be updated by the prototypes of the newly included sample. Hence, the proposed FlexClust clustering algorithm is not only scalable because it works within limit of memory by drawing samples incrementally and compressing the data, but it also has the advantage of clustering open data stream as the algorithm provides usable model at any time.

4.4 Properties of FlexClust Algorithm

The proposed FlexClust clustering algorithm has the following properties.

- **Scalability**: FlexClust is scalable in terms of 1) data size - Incremental compression procedure maintains only the prototype system in the memory and purges the data points to free some memory for filling new data points to the memory buffer, and this makes it scalable to very large data sets, and 2) memory – workable within limited memory even though the data size is too huge to be loaded at once.
- **Exhaustiveness**: All the data points are incrementally included in the clustering process and none goes unused.
- **Recovery of clusters**: The proposed MBF recovers clusters that have been missed out in the initial sample but found in the samples drawn later.
- **Applicability to open data stream**: FlexClust is potentially useful to cluster open data stream where the clusters structure might have changed over time. It

provides up to date usable model by incorporating new arrived data into the current model without recomputing all the previous data.

- **No pre-determination of number of prototypes**: FlexClust employs mixture modelling for data compression which has the advantage of automatically determining the number of clusters that best fit the compressed data.

4.5 Experimental Evaluation

Benchmark Comparison

The performance of the proposed FlexClust is compared to four clustering algorithms for large data set: sufficient EM (Steiner et al., 2007), SPSS TwoStep (SPSS Inc., 2003), model from sample (strategy III) (Wehrens et al., 2004), and CLARA (Kaufman et al., 1990).

Sufficient EM (Steiner et al., 2007) is a two-step procedure where the observation points are first compressed and then clustered. In contrast to the proposed FlexClust, sufficient EM compresses the data set all at once. Sufficient EM employs *k*-means algorithm to compress all the observation points and represents them by the prototypes characterized by sufficient statistics, i.e. means and covariances, and the number of observations in the condensed data set. The clustering step is carried out by Gaussian mixture model where the parameters are estimated from a variant type of EM algorithm.

SPSS TwoStep clustering algorithm (SPSS Inc., 2003) is a two-step procedure which compresses the observation points into prototypes in the first step, and then clusters the resulting prototypes in the second step. It employs data compression procedure similar to the BIRCH algorithm, where data are condensed into cluster features (CF) and summarized into an incremental built CF tree. The cluster feature is characterized by a triple of summarized information which comprises the number of data points in the subcluster, the linear sum and the square sum of the data points. In the second step, the resulting prototypes (or leaf nodes) are clustered using an agglomerative hierarchical clustering method, where only the variance of the compressed data will be considered.

Model from sample (strategy III) (Wehrens et al., 2004) applies a basic model-based clustering to a sample of the data, and then extends several tentative best models from the sample via EM to the whole data in more iterations to eventually select the best model from the tentative best models.

CLARA (Kaufman et al., 1990) trains a model from a random sample and then performs discriminant analysis to classify the rest of the data. The procedure makes it different from the other sampling-based algorithms in that it repeats the draw of random sample for a few times to find a set of mediods that gives the best clustering result with the smallest average distance.

The sufficient EM and SPSS TwoStep are chosen for comparison because this thesis intends to compare the following few aspects: 1) the performance of different compression procedures – the incremental compression from FlexClust versus the one time compression from sufficient EM and the incremental one time compression from SPSS TwoStep, 2) the effectiveness of different compression methods – the mixture model compression from FlexClust versus the *k*-means from sufficient EM and SPSS TwoStep, and 3) the performance of the model selection criterion – the MBF from FlexClust versus the variant of BIC from sufficient EM.

The model from sample (strategy III) and CLARA are chosen in this thesis to compare the aspect of sample representativeness for different sampling schemes – the incremental sampling with the flexibility of clusters recovery from FlexClust versus sampling the best sample from few samples from CLARA and select the best model from few models trained from a sample from strategy III.

Several criteria are used to assess the aforementioned clustering algorithms. One aspect is the accuracy of the clustering result in terms of classification accuracy and estimate of parameters, and the correct number of clusters. Another aspect is the stability of the clustering algorithm over the effects of initial sample, sample size, compression method and compression rate.

Design of Analysis

FlexClust starts by drawing random sample of observations from the data for compression. To investigate the effect of sample size, two different sample sizes were compared. Each sample size was performed 10 times with different initial random sample, so that the conclusions do not depend on the particular sample drawn and the size of the sample being drawn. The incremental compressed samples maintain at a constant size throughout the compression process. To reduce computational load in testing out FlexClust algorithm, the sample sizes considered were relatively smaller than the memory buffer size.

For sufficient EM, the influence of the compression degree on the resulting cluster structure is taken into consideration by setting two different numbers of prototypes. For each prototype system, 10 experiments were carried out by different starting solution for the *k*-means compression step to study the stability of the compression method. Due to irregular trend in the variant of BIC, a range contains the actual number of clusters was considered for each set of simulated data. The number of clusters was determined from the minimum value of the variant of BIC in the given range. The iteration of EM algorithm was limited to a maximum number of 3000 steps. It stopped even though the log-likelihood function is not converged.

In SPSS TwoStep clustering algorithm, the size of CF-tree for comparable setting with the sample size in FlexClust was considered. The depth and level of the CF-tree was set as such that the number of prototypes roughly corresponded to the sample size in FlexClust. For example, the CF-tree was set at a depth of three levels with a maximum of eight branches at each node, 8^3-tree, such that the number of prototypes was restricted to 512 prototypes to correspond to the sample size of 500 in FlexClust. The observation data was randomly sorted to generate 3 replications to cater the possible dependency of CF-tree on the input order of the observation data.

For comparison purpose, the initial sample sizes in strategy III were set equal to FlexClust. Similar to FlexClust, 10 experiments were performed in strategy III for two different sample sizes respectively. Strategy III selected 5 tentative best models based on the training set with consideration for the 10 model parameterizations available in MCLUST (see Table 2.1), and ran at most 100 EM steps to classify the whole data set. The best model was selected from these five.

Since CLARA algorithm applies PAM (partition around medoids) on the best selected sample and gives the best clustering as the output, only 1 run was conducted. The quality of CLARA clustering algorithm is measured based on the average silhouette width.

Software

All experiments in this thesis, unless specified, were performed in the statistical programming environment R (Ihaka and Gentlemen, 1996) version 2.4.1. Strategy III and FlexClust used the MCLUST package (Fraley and Raftery, 1999) version 3.0-0 which considers ten parameterizations of the cluster covariance matrices (see Table 2.1). SPSS TwoStep was carried out in SPSS 12.0.1 for Windows.

4.5.1 Simulation Study 1

The simulated data set consists of 15,000 data points generated from a seven-component two-dimensional Gaussian mixture distribution. The clusters are of different sizes and shapes and some are overlapped as graphically shown in Figure 4.3. Special attention is paid to the relatively small nested cluster 6. For this set of simulated data, the sample sizes considered in FlexClust and strategy III are 500 and 1000, and the numbers of prototypes considered in sufficient EM are 500 and 800. The range of number of clusters considered for sufficient EM is from 2 to 12.

The parameters for the data set are as follows:

(μ_1, Σ_1, n_1) = ((-10,38), (1.5,-1,-1,20), 2000),
(μ_2, Σ_2, n_2) = ((-6,35), (20,0,0,1), 4000),
(μ_3, Σ_3, n_3) = ((7,30), (6,0.5,0.5,3), 3500),
(μ_4, Σ_4, n_4) = ((30,60), (3,0,0,33), 1000),
(μ_5, Σ_5, n_5) = ((-25,35), (8,-0.1,-0.1,1), 2000),
(μ_6, Σ_6, n_6) = ((-29,34), (0.5,0,0,0.5), 1000),
(μ_7, Σ_7, n_7) = ((-50,50), (0.5,3,3,20), 1500).

Results for the number of clusters obtained by FlexClust, sufficient EM, SPSS TwoStep, strategy III and CLARA on the simulated data are shown in Table 4.1. It can be seen that FlexClust outperforms the rest of the algorithms in terms of the chances of obtaining the model that has the correct number of clusters and accurate estimate of parameters. In 10 simulation experiments, FlexClust with the sample size of 500 (i.e. FlexClust [500]) and 1000 (i.e. FlexClust [1000]) are 100% and 50% respectively successful in identifying the correct model, compared to only 30% and 20% respectively for sufficient EM with the number of prototypes 500 (i.e. sufficient EM [500]) and 800 (i.e. sufficient EM [800]). SPSS TwoStep, strategy III and CLARA fail completely in choosing the correct model. SPSS TwoSTep and CLARA underestimate the actual number of clusters whereas strategy III tends to overestimate the actual number of clusters. Both strategy III with the sample size of 500 (i.e. strategy III [500]) and 1000 (i.e. strategy III [1000]) most frequently select models with 10 clusters. This result is consistent with the finding from Wehrens et al. (2004) that strategy III invariably leads to model with more clusters or are more complex than strategy whereby EM is performed on the whole data set for only the best model from the sample.

Table 4.1. Number of clusters and percentages of getting the correct clusters obtained using sufficient EM, SPSS TwoStep, strategy III, CLARA and FlexClust algorithm on the simulated data. (Numbers in the brackets indicate the number of prototypes for sufficient EM algorithm, and the sample size for Strategy III and the proposed FlexClust algorithm).

Algorithm	Number of clusters (Frequency)	% of getting all the correct clusters
Sufficient EM (Steiner et al., 2007) [500]	7*(3), 8(6), 9(1)	30%
Sufficient EM (Steiner et al., 2007) [800]	7*(2), 7(2), 8(3), 9(3)	20%
SPSS TwoStep (SPSS Inc., 2003)	5(3)	0%
Strategy III (Wehrens et al., 2004) [500]	6(2), 9(2), 10(5), 11(1)	0%
Strategy III (Wehrens et al., 2004) [1000]	6(1), 8(1), 9(3), 10(4), 11(1)	0%
CLARA (Kaufman et al., 1990)	5(1)	0%
Proposed FlexClust [500]	7*(10)	100%
Proposed FlexClust [1000]	7*(5), 8(5)	50%

Note: * indicates the correct 7 clusters. s.d. = standard deviation.

In terms of the recovery of the small and nested cluster 6, the MBF criterion in FlexClust performs better than the variant of BIC in sufficient EM and the BIC in strategy III. The small cluster 6 nested in cluster 5 is not identified in most of the initial samples in FlexClust, however it is recovered in the later samples and the proposed MBF criterion suggests correctly that it is a new cluster that has not been identified in the samples before. Figure 4.4 illustrates how the FlexClust algorithm recovered the small nested cluster from incremental compression of samples data in one of the experiments form FlexClust [500]. Figure 4.4 (a) shows the compressed means and covariances of the initial sample of size 500. Apparently, cluster 6 is not found at this stage. In the third sample, cluster 6 is identified as depicted in Figure 4.4 (b), and the MBF criterion suggests that it is a new cluster. For FlexClust [500], from Figures 4.4 (b) to 4.4 (c), we find that no new cluster is found, and the estimates of parameters are getting very close to the true values except for cluster 5. Fifty and thirty percent of the cases in sufficient EM [500] and sufficient EM [800] respectively miss out the nested clusters 6 and identify superfluous components or even identical clusters, for example, see Figure 4.5. The performance of sufficient EM very much depends on how well the one time compression by *k*-means preserves the structure of the clusters. However, for overlapping and nested clusters, *k*-means gives different compression results from different starting seeds and this causes inconsistency in the final models obtained by sufficient EM. For strategy III, 40% and 20% of the experiments in strategy III [500] and strategy III [1000] respectively fail to recover the nested cluster 6 and identify superfluous sparse clusters as shown in Figure 4.6.

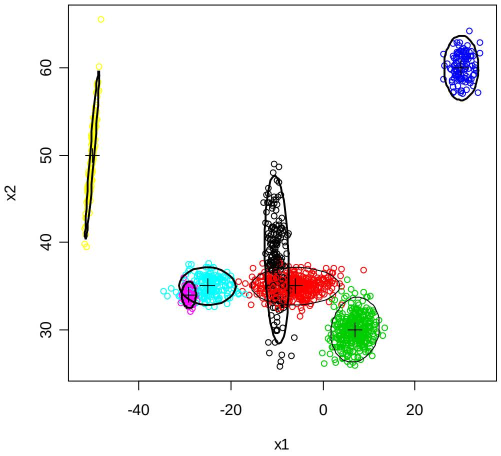
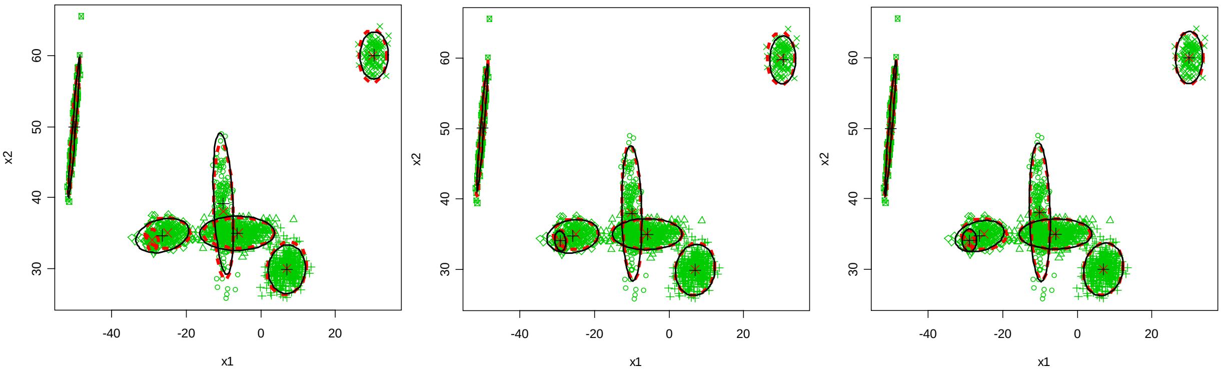
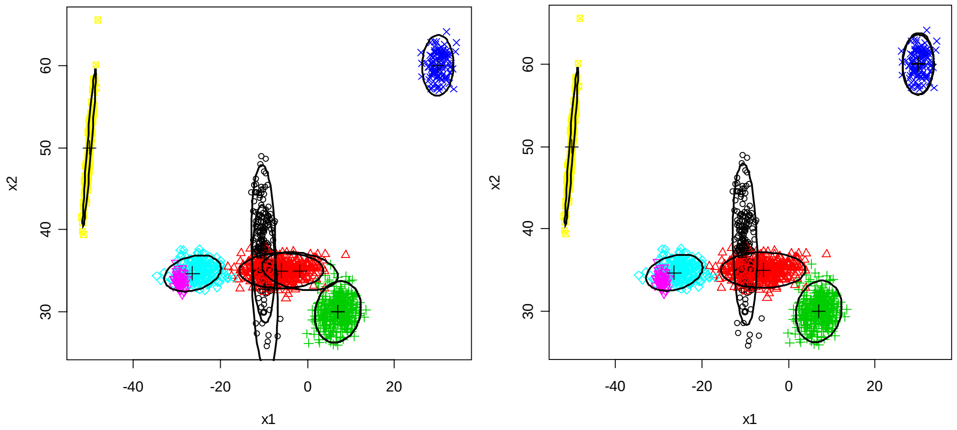


Figure 4.3. True clusters structure (scatterplot of 10% of the total data points) for the data simulated from a seven-component multivariate normal mixture, which has different shape, volume and orientation. (Note: '+' represent means, and covariances are visualized by 90% normal tolerance ellipsoids).



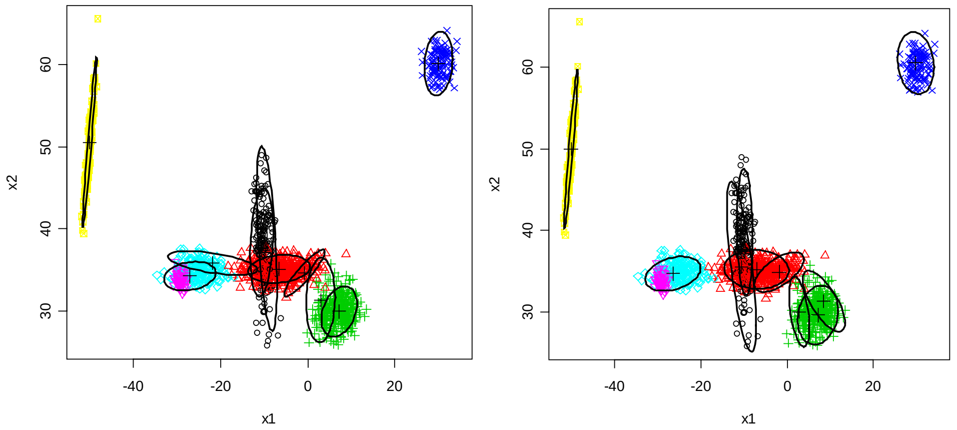
(a) (b) (c)

Figure 4.4.True clusters structure (scatterplot of 10% of the total data points) and the MLE of the true model means and covariances ('x', red dotted line) compared to the current model ('+', black solid line) fitted from incremental compression after: a) initial sample, b) 3 samples, and c) the final sample, in one of the experiments of FlexClust [500]. (Note: covariances are visualized by 90% normal tolerance ellipsoids).



(a) (b)

Figure 4.5. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) sufficient EM [500]-misses out cluster 6 but identifies superfluous clusters at cluster 1 and 2, and b) sufficient EM [800]-misses out cluster 6 but identifies 2 identical cluster 4. (Refer Figure 4.3 for note.)



(a) (b)

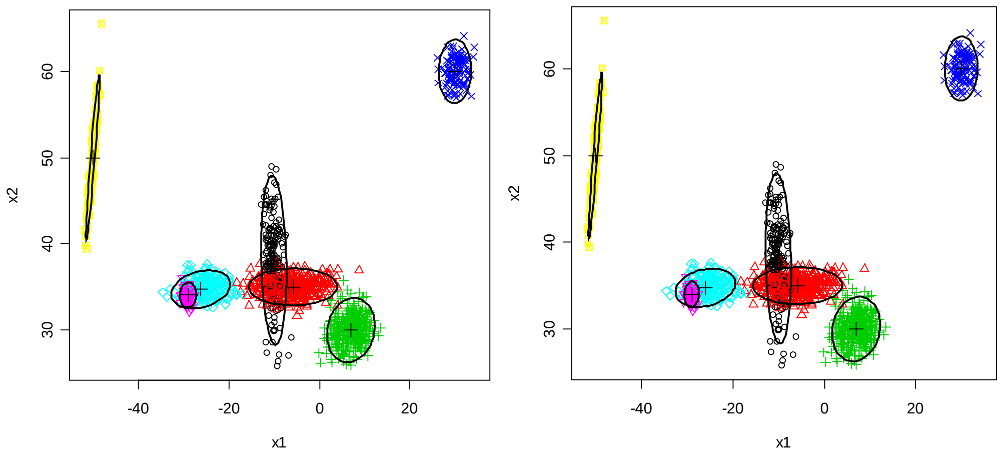
Figure 4.6. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) strategy III [500], and b) strategy III [1000]. Both miss out cluster 6 but identify superfluous clusters with *k*=10. (Refer Figure 4.3 for note.)

Table 4.2. Misclassification rate obtained using sufficient EM, SPSS TwoStep, strategy III, CLARA and FlexClust algorithm on the simulated data. (Refer Figure 4.3 for note.)

Algorithm	Misclassification rate (%)	
	mean	s.d.
Sufficient EM (Steiner et al., 2007) [500]	9.32	3.50
Sufficient EM (Steiner et al., 2007) [800]	6.53	2.19
SPSS TwoStep (SPSS Inc., 2003)	20.79	0.02
Strategy III (Wehrens et al., 2004) [500]	9.28	2.70
Strategy III (Wehrens et al., 2004) [1000]	11.15	3.13
CLARA (Kaufman et al., 1990)	22.59	-
Proposed FlexClust [500]	10.77	0.02
Proposed FlexClust [1000]	9.82	0.97

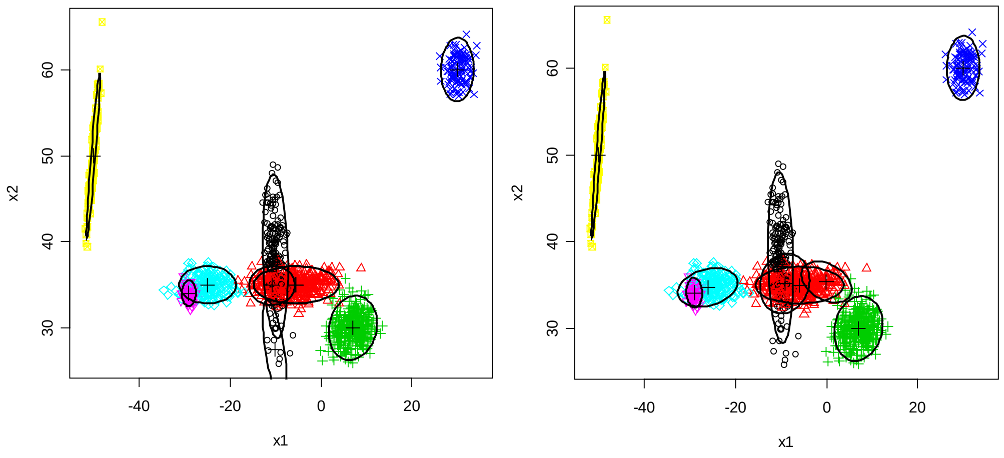
The misclassification rates of simulation study 1 are shown Table 4.2. The misclassification rate of FlexClust are slightly higher than the sufficient EM and strategy III [500] for small sample sizes, but slightly lower than the strategy III [1000] and much lower than the SPSS TwoStep and CLARA. However, the incremental compression through mixture model in FlexClust preserves the structure better compared to the other algorithms. The estimation of model parameters and classification structures on the simulated data for all the five methods are depicted in Figures 4.7 – 4.10. FlexClust [500] classified the data correctly for all the clusters except there is no assignment to cluster 6. FlexClust [500] failed to find the global maximum likelihood estimator particularly for the nested clusters in the algorithm of incremental compression using mixture model and incorporates into the update of the current model. However, when the sample size increases, the MLE of parameters approaches the true values and there is increase in the classification accuracy of FlexClust [1000]. The estimate of parameter means and covariances of the final model on the simulated data for the selected experiments from FlexClust are depicted in Figure 4.7. Thirty and fifty percent of the experiments in sufficient EM [500] and sufficient [800] respectively did not find the global maximum likelihood estimators. The numbers of clusters in the final models of these cases are greater than 7, but the conditional probabilities mapped the observations into the correct 7 clusters, and left some clusters empty, for example see Figure 4.8. Strategy III also suffers from the problem of finding the global maximum likelihood estimators. The maximized a posterior (MAP) in the strategy III do not assign any data points to few of the clusters, and some clusters are sparse, for example see Figure 4.9. For SPSS TwoStep and CLARA, the classification accuracies are respectively 10% and 11.82% lower than FlexClust [500]. SPSS TwoStep employs *k*-means for compression does not preserve the clusters structure well as shown in Figure 4.10 (a). The repeated sampling in CLARA does not draw representative sample to classify the remaining data and obtains clusters structure that cannot distinguish overlapping structure as shown in Figure 4.10 (b).

The FlexClust algorithm especially FlexClust [500] performs consistently in obtaining the final model, which implies that the effect of initial sample is very minimal. Like other sampling-based algorithm, sample size does affect the performance of FlexClust. The complexity in terms of number of clusters of the final model obtained by FlexClust is observed to increase with the sample size. More clusters are used to describe the sample especially at the overlapping area between the elongated cluster 1 and cluster 2 when the sample size is increased. At a fixed compression rate and sample size respectively, sufficient EM and strategy III show higher variability of the number of clusters and clusters structure in the final models. For sufficient EM, the classification accuracy declines at a higher rate of compression. The effect of sample size is not obvious in strategy III.



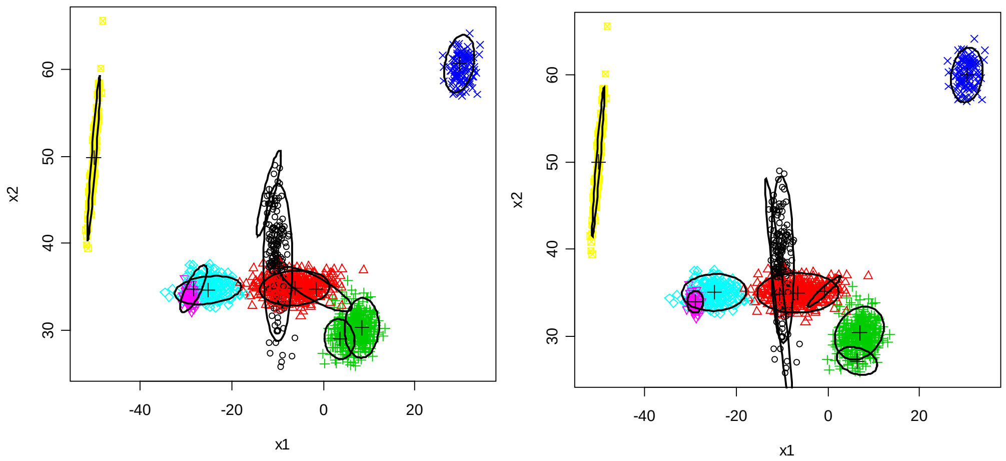
(a) (b)

Figure 4.7. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: (a) FlexClust [500] (k = 7*), (b) FlexClust [1000] (k = 7*). (Refer Figure 4.3 for note.)



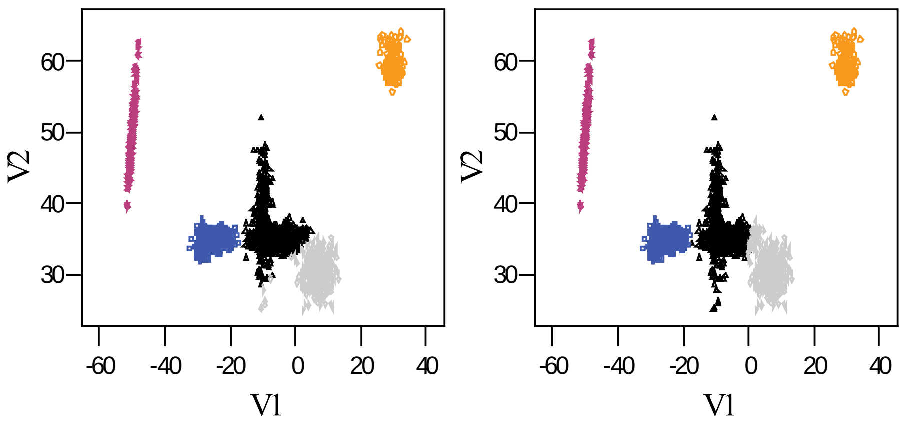
(a) (b)

Figure 4.8. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) sufficient EM [500], and b) sufficient [800]. Non global maximum likelihood estimators and no assignment to some of the clusters. (Refer Figure 4.3 for note.)



(a) (b)

Figure 4.9. True clusters structure (scatterplot of 10% of the total data points), and the MLE of means and covariances obtained using: a) strategy III [500], and b) strategy III [1000]. Non global maximum likelihood estimators. No assignment to one of the 10 clusters and some clusters are sparse. (Refer Figure 4.3 for note.)



(a) (b)

Figure 4.10. Classification structure obtained using: a) SPSS TwoStep ($k = 5$), and b) CLARA ($k = 5$). Plots use 10% of the total observations.

4.5.2 Simulation Study 2

A sample of size 20,000 was generated from a three-component thirteen-dimensional normal mixture. The population parameters are obtained by fitting the *wine* data set from the UCI machine learning repository (Asuncion & Newman, 2007) into a mixture of three-component VVI (see section 2.1) model. The *wine* data set is concerned with the chemical quantities of 13 constituents found in each of the three types of wines grown in the same region in Italy. It has "well behaved" class structures. The mixture proportions of the data are

$$\pi_1 = 0.3178592, \ \pi_2 = 0.2868950, \text{ and } \pi_3 = 0.3952458.$$

The component means are given as follows

$$\mu_1 = (13.7717075, 1.9638926, 2.4404059, 16.8608302, 106.0323937, 2.8509812, 2.9991609, 0.2862978, 1.9027869, 5.5925554, 1.0646513, 3.1570950, 1130.2923859),$$

$$\mu_2 = (13.1261529, 3.2768523, 2.4212224, 21.3087811, 99.0211310, 1.6750952, 0.8154962, 0.4509664, 1.1562232, 7.2203250, 0.6954193, 1.6927530, 627.2638145),$$

$$\mu_3 = (12.289382, 1.953201, 2.267386, 20.296714, 95.205399, 2.298128, 2.130312, 0.357933, 1.655592, 3.058779, 1.061436, 2.840084, 525.396082).$$

The component covariance matrices are all diagonal and given by

$$\Sigma_1 = \text{diag}(0.2007727, 0.4026683, 0.04164438, 5.420391, 107.5776, 0.1149230, 0.1554843, 0.004382626, 0.1732011, 1.466652, 0.01298471, 0.1289701, 44988.93),$$

$$\Sigma_2 = \mathrm{diag}(0.2762789, 1.218337, 0.03638959, 5.255569, 115.1763, 0.1256041, 0.1004483, 0.01492667, 0.1658363, 5.472534, 0.01495655, 0.07154705, 14307.66),$$

$$\Sigma_3 = \mathrm{diag}(0.3006426, 1.031226, 0.1130606, 11.27366, 290.3114, 0.2743027, 0.4760694, 0.01423743, 0.3449565, 0.7489613, 0.04024906, 0.2082159, 25260.28).$$

The sample sizes considered for FlexClust and the strategy III are 250 and 500. The numbers of prototypes for the sufficient EM are set at 500 and 1000. The range of number of clusters considered for the sufficient EM is between 2 to 10.

Table 4.3. Misclassification rate and numbers of clusters using sufficient EM, SPSS TwoStep, strategy III, CLARA, and FlexClust on the data generated from *wine* data set. See caption of Table 4.1.

Method	Number of clusters				Misclassification rate (%)	
	mean	sd	min	max	mean	sd
Sufficient EM (Steiner et al., 2007) [500]	3.5	1.51	2	6	59.32	2.64
Sufficient EM (Steiner et al., 2007) [1000]	2.8	0.79	2	4	42.73	6.57
SPSS TwoStep (SPSS Inc., 2003)	3	0	3	3	0.26	0.005
Strategy III (Wehrens et al., 2004) [250]	4.1	0.57	3	5	1.17	2.47
Strategy III (Wehrens et al., 2004) [500]	4.9	0.32	4	5	3.07	3.80
CLARA (Kaufman et al., 1990)	2	-	-	-	32.43	-
Proposed FlexClust [250]	3	0	3	3	0.15	0.0046
Proposed FlexClust [500]	3	0	3	3	0.15	0.0025

Results of the five methods on the simulated data from the *wine* data set are shown in Table 4.2. The proposed FlexClust identifies the final model with correct number of clusters for 10 out 10 different initial samples of the sizes 250 and 500 respectively. In fact, all the initial samples of sizes 250 and 500 manage to identity the embedded 3 clusters, and throughout the incremental compression of samples, there is no new cluster being added. All the 3 clustering solutions from SPSS TwoStep identify the correct number of clusters and show consistent misclassification rate. This is not surprising as it is mainly due to the population covariance matrices which are all diagonal. FlexClust has the lowest misclassification rate among all the methods. The misclassification rate for SPSS TwoStep is 0.11% lower than FlexClust. Strategy III identifies the correct number of clusters only once when the sample size is 250 and not when the sample size is increased to 500. The misclassification rate in strategy III [250] and strategy [500] are 1.02% and 2.92% respectively higher than FlexClust with the equal sample sizes. The misclassification rates for sufficient EM[500] and sufficient EM[1000] are 59.32% and 43.72% respectively. These are far higher than the rest of the methods for the simulated high dimensional data. Thirty percent of the experiments in sufficient EM [500] and forty percent of the sufficient EM [1000] choose the model with 3 clusters, but the MLE of model parameters do not converge to the global maximal. The average misclassification rate accuracies for these three-component models are up to 60.34% and 40.31% respectively for the sufficient EM [500] and sufficient EM [1000]. CLARA chooses a model with underestimated number of clusters and the misclassification rate is 32.43%.

4.5.3 Simulation Study 3

In simulation study 3, a sample consisting of 10,000 simulated points was generated from a three-component four-dimensional normal mixture model. The population parameters are from the well known *iris* data available at UCI machine learning repository (Asuncion & Newman, 2007) website. Two of the three classes (Versicolor, Virginica, and Setosa) in the *iris* data are overlapping. The component means of the best fitted three-component model are

$$\mu_1 = (5.006, 3.428, 1.462, 0.246),\ \ \mu_2 = (5.914879, 2.777504, 4.203758, 1.298819),$$
$$\mu_3 = (6.546670, 2.949495, 5.481901, 1.985322),$$

and the component covariance matrices are

$$\Sigma_1 = \begin{pmatrix} 0.13322911 & 0.10940214 & 0.01919601 & 0.01158793 \\ 0.10940214 & 0.15497824 & 0.12098300 & 0.01001168 \\ 0.01919601 & 0.12098300 & 0.018176976 & 0.005819438 \\ 0.01158793 & 0.01001168 & 0.005819438 & 0.010693650 \end{pmatrix},$$

$$\Sigma_2 = \begin{pmatrix} 0.22561867 & 0.07613421 & 0.14679059 & 0.04331622 \\ 0.07613421 & 0.08020281 & 0.07370230 & 0.03435134 \\ 0.14679059 & 0.07370230 & 0.16601076 & 0.04947014 \\ 0.04331622 & 0.03435134 & 0.04947014 & 0.03335458 \end{pmatrix},$$

$$\Sigma_3 = \begin{pmatrix} 0.42946303 & 0.10788462 & 0.33465810 & 0.06547643 \\ 0.10788462 & 0.116022293 & 0.08918583 & 0.06141314 \\ 0.33465810 & 0.08918583 & 0.36451484 & 0.08724485 \\ 0.06547643 & 0.06141314 & 0.08724485 & 0.08671670 \end{pmatrix},$$

The mixing proportions are

$\pi_1 = 0.3333333,\ \pi_2 = 0.3003844$, and $\pi_3 = 0.3662823.$

In this simulation study, the sample sizes considered for FlexClust and the strategy III are 200 and 500. For the sufficient EM, the numbers of prototypes considered are 200 and 500, and the range of number of clusters considered is from 2 to 10.

Table 4.4. Misclassification rate and numbers of clusters using sufficient EM, SPSS TwoStep, strategy III, CLARA, and FlexClust on the data generated from *iris* data set. See caption of Table 4.1.

Method	Number of clusters				Misclassification rate (%)	
	mean	sd	min	max	mean	sd
Sufficient EM (Steiner et al., 2007) [200]	3	0	3	3	2.04	0.03
Sufficient EM (Steiner et al., 2007) [500]	3	0	3	3	2.00	0.04
SPSS TwoStep (SPSS Inc., 2003)	2	0	2	2	30.00	0
Strategy III (Wehrens et al., 2004) [200]	3.8	0.42	3	4	4.62	5.70
Strategy III (Wehrens et al., 2004) [500]	4.5	0.07	3	5	6.85	5.59
CLARA (Kaufman et al., 1990)	2	-	-	-	30.01	-
Proposed FlexClust [200]	3	0	3	3	1.91	0.15
Proposed FlexClust [500]	3	0	3	3	1.83	$5.8\text{x}10^{-3}$

Results of the five compared methods on the simulation study 3 are summarized in Table 4.3. The proposed FlexClust and the sufficient EM identify model with the correct number of clusters in all the experiments. However, the misclassification rate of the proposed FlexClust is lower than the sufficient EM by 0.13% and 0.17% respectively when the sample sizes or number of prototypes are 200 and 500. This implies that the parameters estimated in FlexClust are closer to the true value compared to sufficient EM when both methods select the correct number of clusters. For strategy III, only 20% and 10% of the experiments from strategy III [200] and strategy [500] respectively identify the correct number of clusters. The misclassification rates in strategy III are higher than FlexClust by 2.71% and 5.02% respectively for sample sizes 200 and 500. CLARA and SPSS TwoStep give clustering solution with only 2 clusters. It shows that the two methods could not recover the overlapping cluster structure of the data set.

4.6 Application to Real Data

The performance of the proposed FlexClust is compared to sufficient EM and strategy III algorithm in the study of two sets of real data. The first data set is the St Paulia data which has 81,472 pixels. It is an RGB image with 268 columns and 304 rows. The data set is available at www.cac.science.ru.nl/people/ rwehrens/publications.html. The RGB image is shown in Figure 4.11. Identifying the small yellow flowers is of particular interest in this study. The second data set is the Forest CoverType data. It describes forest cover type from cartographic variables (no remotely sensed data), which were derived from data originally obtained from US Geological Survey (USGS) and US Forest Service (USFS) data. The data has 581,012 data items and available at UCI machine learning repository (Asuncion & Newman, 2007). Five quantitative attributes as used in Jin et al. (2004) are considered in this thesis. The sample sizes considered in FlexClust and the strategy III are 1000 and 2000, and the numbers of prototypes considered in sufficient EM are 1000 and 2000. The range of number of clusters considered for sufficient EM is from 2 to 35 for the St Paulia data and from 2 to 14 for the Forest CoverType data. For St Paulia data, the performances of the three algorithms are assessed visually, whereas for the Forest CoverType data, the performances are assessed by comparing the average log-likelihood of the obtained Gaussian mixture. For each setting of the strategy III and FlexClust algorithms, 10 experiments were carried out for different initial samples. For each prototype system in the sufficient EM, 10 replications were carried out by considering different starting solution for the *k*-means compression.

Results for St Paulia Data

Results for sufficient EM, strategy III and the proposed FlexClust algorithms on the St Paulia image data are summarized in Table 4.4. FlexClust is more stable than sufficient EM in choosing the final model. The scan through and select procedure in FlexClust drew samples in blocks of size 2000 and identified a set of 120 data points from the small clusters with proportion less than 0.01. Most of the points in this set were the pixels for the yellow flowers. For sufficient EM, some of the experiments obtained clusters with mixture proportions as low as $1\text{x}10^{-71}$ – $1\text{x}10^{-5}$. It implies that FlexClust overcomes the problem of missing out on small clusters in the incremental compression procedure and performs as good as the one time compression where data set is compressed as a whole. Examples for the segmentation results from each algorithm are shown in Figure 4.12. The segmentations from both the FlexClust and the sufficient EM algorithms reveal the yellow flowers in all the experiments. The pixels of the yellow flowers are from small clusters but they do not overlapped with other clusters and are not nested in any clusters, therefore, *k*-means works reasonably well to identify them as clusters either in the scan through and select procedure or the compression step at sufficient EM. All the experiments from the strategy III missed out on the small clusters of yellow flowers.

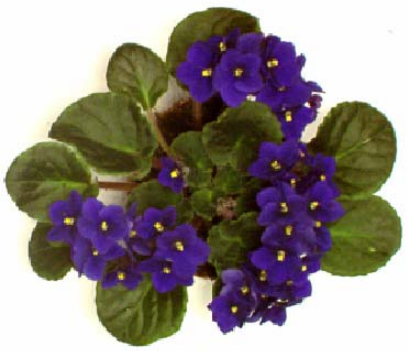
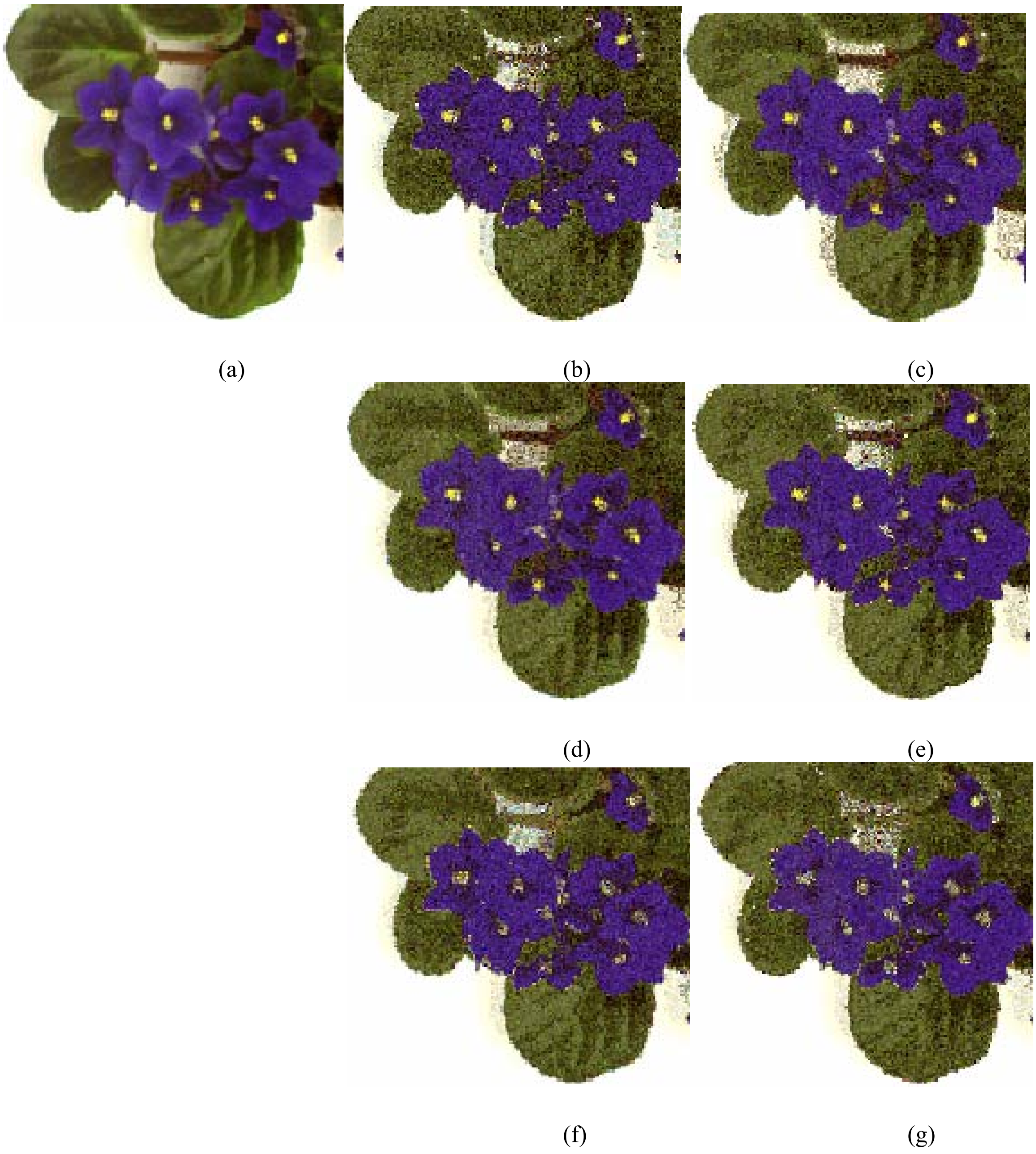


Figure 4.11. RGB image of the St Paulia data set



(a) (b) (c)

(d) (e)

(f) (g)

Figure 4.12. Ground true image on St Paulia RGB image data in (a). Segmentation results on the image data obtained by: b) FlexClust [1000] (*k* = 24), c) FlexClust [2000] (*k* = 34), d) sufficient EM [1000], (*k* = 26), e) sufficient EM [2000] (*k* = 27), f) strategy III [1000] ((*k* = 11), and g) strategy III [2000] (*k* = 10).

Table 4.5. Number of clusters obtained using the sufficient EM, strategy III and FlexClust on the St Paulia data. See caption of Table 4.1.

Algorithm	Number of clusters, *k*			
	mean	sd	min	max
Sufficient EM (Steiner et al., 2007) [1000]	27	2.3	24	31
Sufficient EM (Steiner et al., 2007) [2000]	27.2	5.5	20	35
Strategy III (Wehrens et al., 2004) [1000]	9.7	1.2	8	11
Strategy III (Wehrens et al., 2004) [2000]	9.6	1.5	7	13
Proposed FlexClust [1000]	25.1	1.5	23	27
Proposed FlexClust [2000]	34.3	3.3	30	38

Results for Forest CoverType Data

The average log-likelihood values of FlexClust, sufficient EM and strategy III are –786,687, -809,407 and –853,212 respectively (note: 21,000,000 was added to all the log-likelihood values for legibility). Thus, FlexClust generates slightly more accurate Gaussian Mixture than both sufficient EM and strategy III.

4.7 Very Large Simulated Data

A data of size 2 million was generated from a seven-component five-dimensional normal mixture. In the mixture distribution, component one and two, and also component three and four, share common mean but have different covariances. The parameters are

$\pi_1 = \pi_2 = \pi_4 = \pi_6 = 0.15$, $\pi_3 = 0.175$, $\pi_5 = 0.125$, $\pi_7 = 0.1$,

$\mu_1 = \mu_2 = (0,0,0,0,0)$, $\mu_3 = \mu_4 = (4,4,4,4,4)$, $\mu_5 = (4,4,4,4,-4)$, $\mu_6 = (4,-4,-4,4,4)$,

$\mu_7 = (-4,4,4,-4,4)$, and $\Sigma_1 = \Sigma_3 = \Sigma_5 = \Sigma_6 = \Sigma_7 = I_5$, $\Sigma_2 = \Sigma_4 = 4I_5$.

The whole data set is too large to be loaded into the memory buffer. It gives two implications: 1) the pre-clustering compression step as in the sufficient EM and SPSS TwoStage could not be carried out, and 2) for the strategy III and CLARA, the samples drawn from the portion of data set that loaded into the memory buffer are not representative enough and might miss out on important clusters.

For FlexClust, the incremental compression procedure works scalable to the data size. FlexClust [1000] and FlexClust [2000] identify the number of clusters correctly and the classification accuracies are 90.33% and 90.22% respectively.

4.8 Discussion and Conclusion

The proposed novel FlexClust clustering algorithm has been tested over simulated data and real life data. Compared to the sufficient EM, SPSS TwoStep, sample from model (strategy III), and CLARA, the results obtained from FlexClust are promising.

In terms of compression-based clustering algorithms, FlexClust shows two main advantages over the sufficient EM and SPSS TwoStep algorithms. Firstly, FlexClust reduces the loss of information. This is mainly due to a more effective prototype system in representing the compressed data, and a more efficient model for the incorporation of incrementally compressed data. FlexClust incorporates the incrementally compressed information by mixture model into the current model has given estimate of model parameters that are close to the true values. The compression method by Gaussian mixture model and representation of the summarized information using the prototype of MLEs of mixture model preserves the clusters structure well. On the contrary, the one time compression in sufficient EM without the guideline on the sufficient number of prototypes very much depends on the compression quality by *k*-means. Table 4.6 shows the misclassification rates for one-time compression and incremental compression for data sets with dimensions of 2, 4, and 13 as in simulation study 1, 2 and 3. For the two Gaussian mixture models, the mean misclassification rate of sufficient EM is 50.88% higher than FlexClust in the 13-dimensional data as shown in Figure 4.13. This is mainly caused by the inefficiency of *k*-means for high dimensional data. For overlapped and nested clusters, *k*-means gives different compression results from different starting seeds and this causes inconsistency in the final models obtained by sufficient EM. Most of the time, sufficient EM converges poorly and selects a less parsimonious model with some empty clusters as the best model. SPSS TwoStep characterizes the dispersion of the compression data by its variance instead of the full covariance has worsened the loss of information due to compression. Secondly, FlexClust is scalable to any data size. In the case of the data set size larger than the memory buffer, the one time compression procedure in sufficient EM and SPSS TwoStep is not applicable whereas the incremental compression procedure in FlexClust performs satisfactorily even for data with small clusters or overlapping clusters.

Table 4.6. Misclassification rates for one-time compression and incremental compression methods for data sets with different dimensions.

Algorithm	Two-dimensional Misclassification rate (%)		Four-dimensional Misclassification rate (%)		Thirteen-dimensional Misclassification rate (%)	
	mean	s.d.	mean	s.d.	mean	s.d.
One-time compression:						
Sufficient EM (Steiner et al., 2007) [s1]	9.32	3.50	2.04	0.03	59.32	2.64
Sufficient EM (Steiner et al., 2007) [s2]	6.53	2.19	2.00	0.04	42.73	6.57
SPSS TwoStep (SPSS Inc., 2003)	20.79	0.02	30.00	0	0.26	0.005
Incremental compression:						
Proposed FlexClust [s3]	10.77	0.02	1.91	0.15	0.15	0.0046
Proposed FlexClust [s4]	9.82	0.97	1.83	$5.8\text{x}10^{-3}$	0.15	0.0025
Note:	s1=500, s2=800, s3=500, s4=1000.		s1=s3 =200, s2=s4=500.		s1=500, s2=1000, s3=250, s4=500.	

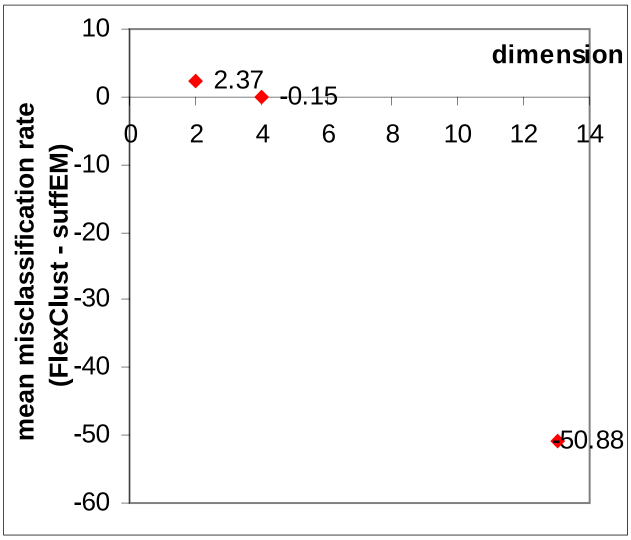


Figure 4.13. Difference of mean misclassification rates between FlexClust and sufficient EM.

In terms of sampling-based clustering algorithms, FlexClust outperforms the strategy III and CLARA algorithms by obtaining models that produce lower misclassification rates for data sets with different dimensions from simulation study 1, 2 and 3 as shown in Table 4.7. The scheme of incremental random sampling in FlexClust overcomes the ill sample problem faced in the sampling schemes of strategy III and CLARA. FlexClust is insensitive to initial sample being drawn and shows more consistent and accurate results in determining the number of clusters at a fixed sample size. The problem of ill sample is not obvious in the algorithm as samples are drawn incremental with the flexibility of recovering clusters that have been missed out before. Furthermore, the scan through and select procedure helps to identify data points from probable small clusters. However, the sample size does affect the results of FlexClust. Although CLARA is proposed to improve the sampling-based clustering method by drawing few samples, it is inefficient to select the best sample that consists of medoids close to the population medoids. Strategy III trains few tentative best models from a sample does not help to select the model with the correct number of clusters. Most of the time, the strategy converges to bad local minima and selects a less parsimonious model with some empty clusters as the best model. For the two Gaussian mixture models considered, that is FlexClust and strategy III, Figure 4.14 shows that the mean misclassification rate of FlexClust is 0.08-3.87% lower than strategy III.

Table 4.7. Misclassification rates for incremental sampling and random sampling for data sets with different dimensions.

Algorithm	Two-dimensional		Four-dimensional		Thirteen-dimensional	
	Misclassification rate (%)		Misclassification rate (%)		Misclassification rate (%)	
	mean	s.d.	mean	s.d.	mean	s.d.
Random sampling						
Strategy III (Wehrens et al., 2004) [s1]	9.28	2.70	4.62	5.70	1.17	2.47
Strategy III (Wehrens et al., 2004) [s2]	11.15	3.13	6.85	5.59	3.07	3.80
CLARA (Kaufman et al., 1990)	22.59	-	30.01	-	32.43	-
Incremental sampling						
Proposed FlexClust [s3]	10.77	0.02	1.91	0.15	0.15	0.0046
Proposed FlexClust [s4]	9.82	0.97	1.83	5.8x10^{-3}	0.15	0.0025
Note:	s1=s3=500, s2=s4=1000.		s1=s3 =200, s2=s4=500.		s1=s3=250, s2=s4=500.	

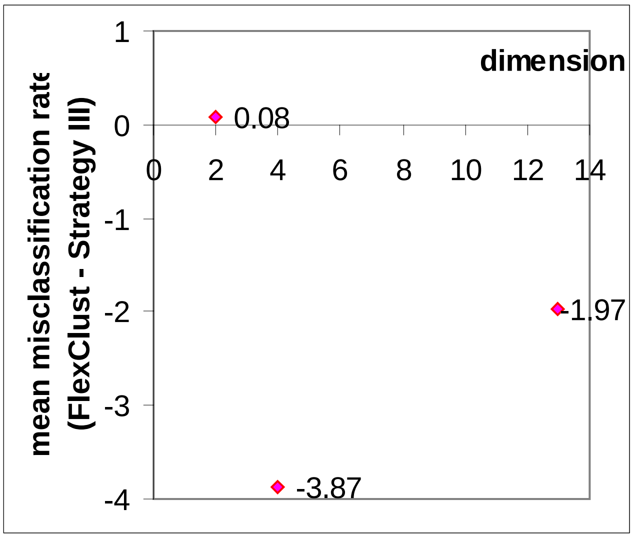


Figure 4.14. Difference of mean misclassification rates between FlexClust and strategy III.

Determining the number of clusters in data is a difficult problem. In dealing with large data clustering, this problem becomes even more challenging. The proposed FlexClust clustering algorithm provides an alternative solution with also taking the memory size into consideration. On top of that, FlexClust is potentially useful to cluster open data stream where the clusters structure might have changed over time. It provides up to date usable model by incorporating new arrived data into the current model without recomputing all the previous data. This issue will be addressed elsewhere. On the contrary, the variant of BIC in sufficient EM always suggests less parsimonious models. The values for the variant of BIC for a range of number of clusters, *G*, can be in an irregular trend, and the number of clusters for the condensed data is determined based on the minimum value for the given range of *G*. It chooses incorrect model if the given *G* does not consist of the true number of clusters. Figure 4.15 shows the percentages of experiments that are able to identify the correct number of clusters and correct clusters in simulation 1, 2 and 3. FlexClust outperforms the rest of the methods in all the data sets. On average, FlexClust identifies the correct clusters or correct number of clusters 38.33%, 58.33%, 85% and 92% of times better than sufficient EM, SPSS TwoStep, strategy III and CLARA respectively.

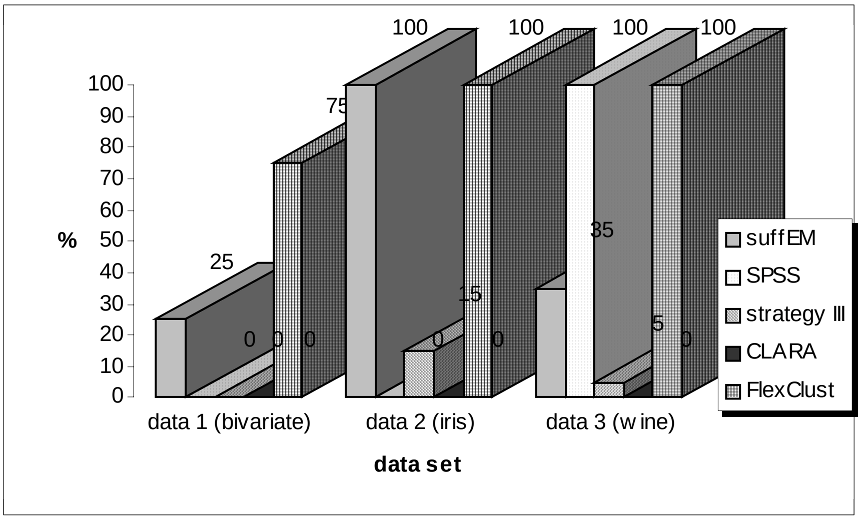
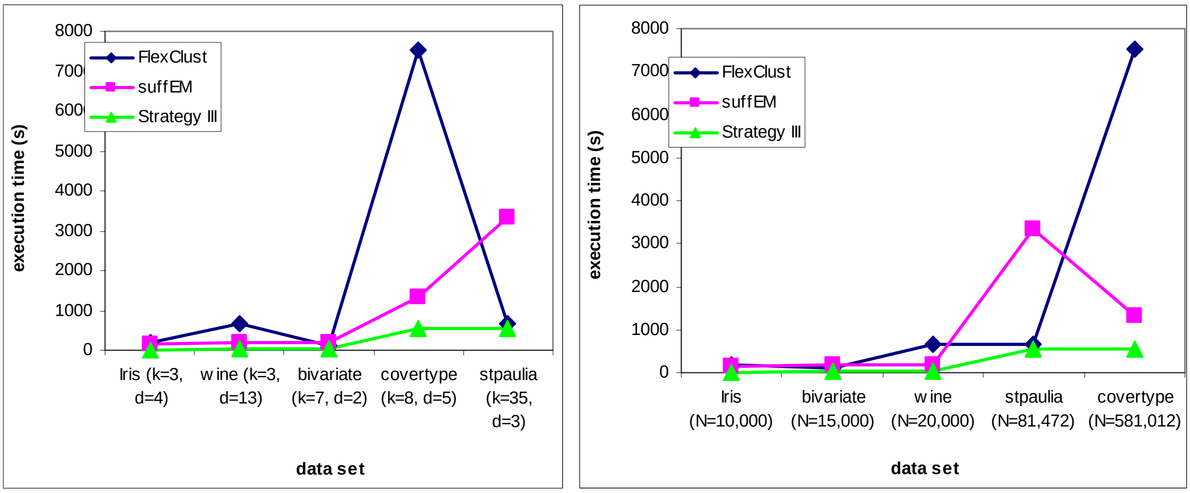


Figure 4.15. Percentages of experiments that are able to identify the correct number of clusters and correct clusters.

The execution time of FlexClust, sufficient EM and strategy III for the synthetic and real data sets are shown in Figure 4.16. The execution time of sufficient EM increases from 146.92 to 3338.66 s when the number of clusters of the data sets increases from 3 to 35 as plotted in Figure 4.16(a). The execution time of FlexClust increases from 100.27 to 7528.89 s when data size increases from 15,000 to 581,012 as shown in Figure 4.16(b). FlexClust runs 5.07 times faster than sufficient EM in St Paulia data with the highest number of clusters, but 5.67 times slower than sufficient EM in Forest CoverType data with the largest data size. Strategy III takes the lowest execution time in all sets of data. It runs 1.2 times and 13.82 times faster than FlexClust in the St Paulia and Forest CoverType data respectively.



(a) **(b)**

Figure 4.16. Performance of average execution time according to (a) number of clusters, (b) data size.

Direct application of mixture model clustering to large data sets is often constrained by three main resources: data size, memory and time. Extensions of the clustering method to large data sets are usually confined to large data sets that can be processed as a whole. However, this is not practical in data mining applications. Incremental data compression according to the available memory buffer and incorporating the compressed information into the current model fit is a solution to this problem. However, this approach faces two challenges. Firstly, the behaviour of the incremental scheme can be viewed as a generalized version of a sampling based scheme, and therefore it suffers from the same shortcomings as other sampling schemes such as non representative sample and missing out on small clusters. Secondly, data compression causes loss of information from the ineffective resulting prototype system. FlexClust algorithm is proposed to address the problems. Results on the simulated and real data support that the proposed FlexClust algorithm is scalable to very large data sets and at the same time overcomes the problems of loss of information due to partition of data and data compression.

5

Scalable Mixture Model Clustering Algorithm for Mixed Data

This chapter consists of two main parts. First, a parametric model for mixed variables is developed and its performance is compared to some existing mixture models for mixed data such as the conditional Gaussian model, restricted location mixture model and modified location model. The developed model is then incorporated in the scalable algorithm proposed in Chapter 4 to introduce a scalable clustering algorithm for mixed data.

5.1 Conditional Gaussian Model

When observations are made on both categorical and continuous variables, the data are said to be mixed-mode or mixed. The assumption of Gaussian mixture model in Chapter 1 for this kind of data is not realistic. Lawrence and Krazanowski (1996) proposed a finite mixture model for the problem of mixed-mode data classification, which has been termed as the conditional Gaussian distribution, or location model. It specifies the joint distribution of mixed-mode data as the product of the marginal distribution of the categorical variables and the conditional distribution of the continuous variables given the values of categorical variables. It is assumed that the continuous variables have a different multivariate normal distribution for each possible setting of categorical variable values, while the categorical variables have an arbitrary marginal multinomial distribution. Given a data set measure on the *i*-th observation of *n* units, $\mathbf{y}_i = (\mathbf{u}_i, \mathbf{x}_i)$, be the mixed variables of q categorical variables, $\mathbf{u}_i = (u_{1i},\ldots,u_{qi})$, and p continuous variables, $\mathbf{x}_i = (x_{1i},\ldots,x_{pi})$. Suppose that the j-th categorical variable has c_j categories, the q categorical variables can be uniquely transformed to a single multinomial random variable W with m cells, w_s (s = 1, …, m), where $m = \prod_{j=1}^{q} c_j$ is the number of distinct combination (location) of the q categorical variables. The associations among the original q categorical variables are then converted into relationship among the resulting multinomial cell probabilities. The given data according to the cell of w_s in which the observation occupies is then denoted by $\mathbf{x} = (\mathbf{x}'_{11},\ldots,\mathbf{x}'_{1n_1},\mathbf{x}'_{21},\ldots,\mathbf{x}'_{2n_2},\ldots,\mathbf{x}'_{m1},\ldots,\mathbf{x}'_{mn_m})'$, where $\mathbf{x}_{si}$ is a p x 1 vector of continuous variables for the i-th out of the n_s observation at location s (s = 1, …, m; $\sum_{s=1}^{m} n_s = n$). The conditional Gaussian model assumes that given the multinomial cell w_s where the observation is placed, the distribution of the continuous variables is multivariate normal, $\mathbf{x}_{si} \sim N(\mu_s, \Sigma_s)$, and that the probability of an observation in the multinomial cell w_s is p_s (s = 1, …, m). In the mixture separation application, Lawrence and Krzanowski (1996) assumed that each observation is drawn from a mixture of g subpopulations with unknown proportions α_k (k = 1, …, g; $\sum_{k=1}^{g} \alpha_k = 1$). The p.d.f. of each observation $\mathbf{x}_{si}$ is given by

$$f(\mathbf{x}_{si}, w_s; \mathbf{\Theta}) = \sum_{k=1}^{g} \alpha_k p_{ks} \phi(\mathbf{x}_{si}; \mu_{ks,} \Sigma_{ks} \mid W = w_s) \qquad (5.1)$$

where $\phi(\mathbf{x}_{si}; \mu_{ks,} \Sigma_{ks} \mid W = w_s)$ is the conditional p.d.f. of multivariate normal for the vector $\mathbf{x}_{si}$ given that it is placed in cell w_s, and p_{ks} is the probability of an observation in cell w_s of the multinomial variable in subpopulation k (s = 1, …, m; k = 1, …, g).

In deriving the distance between populations of mixed data for discriminant analysis using the above model, Krazanowski (1993) pointed out that the large number of parameters contained in the conditional Gaussian model causes estimation problems in many practical situations. To make satisfactory progress, Lawrence and Krazanowski (1996) adopted the idea from the homogeneous conditional Gaussian model in graphical modelling to constrain all the dispersion matrices to be equal, that is, to set Σ_{sk} equal to Σ for all k and s. Actually, the homogeneous conditional Gaussian model was originated by Olkin and Tate (1961) as the location model for discriminant analysis of mixed data. Thus, the conditional Gaussian model for mixture separation by Lawrence and Krzanowski (1996) is sometime termed as the location model.

Structurally the log-likelihood of the conditional Gaussian model is the same as the one for Gaussian mixture model for continuous variables in equation (1.5). Indeed, the similarity of structure can be emphasized by viewing the log-likelihood as a mixture of g x m normal clusters having the hierarchical structure of g clusters and m subclusters within each of these clusters. Then the log-likelihood of the mixture can be written

$$L(\mathbf{\Theta}) = \sum_{k=1}^{g}\sum_{s=1}^{m}\sum_{i=1}^{n_s} z_{ksi}\{\ln \alpha_k + \ln p_{ks} + h(\mathbf{x}_{si};\mu_{ks},\Sigma \mid W = w_s)\} \qquad (5.2)$$

The EM algorithm from section 1.1.1 can be applied to this log-likelihood; it just has an extra summation over the cells, w_s, and extra parameters, p_{ks}. The estimates obtained are

$$\hat{\alpha}_k = \sum\nolimits_{s=1}^{m}\sum\nolimits_{i=1}^{n_s}\frac{\hat{z}_{ksi}}{n} \qquad (5.3a)$$

$$\hat{p}_{ks} = \sum\nolimits_{i=1}^{n_s}\frac{\hat{z}_{ksi}}{n\hat{\alpha}_k} \qquad (5.3b)$$

$$\hat{\mu}_{ks} = \sum\nolimits_{i=1}^{n_s} \frac{\hat{z}_{ksi}\mathbf{x}_{si}}{n\hat{p}_{ks}\hat{\alpha}_k} \tag{5.3c}$$

$$\hat{\Sigma} = \sum\nolimits_{k=1}^{g}\sum\nolimits_{s=1}^{m}\sum\nolimits_{i=1}^{n_s} \frac{\hat{z}_{ksi}(\mathbf{x}_{si} - \hat{\mu}_{ks})(\mathbf{x}_{si} - \hat{\mu}_{ks})'}{n} \tag{5.3d}$$

where

$$z_{ksi} = \frac{\alpha_k p_{ks} \exp\left\{-\frac{1}{2}(\mathbf{x}_{si} - \mu_{ks})'\Sigma^{-1}(\mathbf{x}_{si} - \mu_{ks})\right\}}{\sum\nolimits_{k=1}^{g} \alpha_k p_{ks} \exp\left\{-\frac{1}{2}(\mathbf{x}_{si} - \mu_{ks})'\Sigma^{-1}(\mathbf{x}_{si} - \mu_{ks})\right\}} \tag{5.3e}$$

One of the most fundamental problems for the conditional Gaussian model is non-identifiability. It arises from the indeterminacy of cluster label at each location. In the conditional Gaussian model, locations of the clusters are known and labeled, but cluster labels within the locations are unknown. For the case of *m* locations and *g* clusters, if the cluster labels at the first location are not permuted to avoid obtaining redundant parameters sets, and only the cluster labels in the remaining *m*-1 locations are permuted, there are $(g!)^{m\text{-}1}$ ways of assigning the cluster labels. These different labeling offer different views of cluster structure of the data, but provide the same likelihood. Lawrence and Krzanowski (1996) always choose the labeling that yielded the fewest misclassifications. However, this causes excessive shrinkage of parameter estimates. Willse and Boik (1999) suggested that perhaps the best solution for the problem is to carry out separate cluster analysis within each location, and then expert knowledge is used to assign group labels within locations. They modified the conditional Gaussian model by imposing restrictions on the conditional means of the continuous variables in order to obtained identifiable finite mixture models. The identifiability restriction in the restricted conditional Gaussian model or identifiable location mixture model proposed by Willse and Boik (1999) considered an additive model to include the differences in the conditional means across locations. However, the choice of an identifiability restriction itself is a problem to be solved. Celeux, Hurn and Robert (2000) demonstrated that different restrictions may generate markedly different results. Stephens (2000) showed that many choices of identifiability restriction do not completely remove the non-identifiability problem. This thesis proposes an alternative to solve the problem.

The conditional Gaussian model as originally proposed by Lawrence and Krzanowski (1996) assumes that all the *m* x *g* cells formed from the combination of *m* values of the multinomial variable and the *g* clusters always contain observations. However, in practical applications, it is very likely that some of the cells may be empty, and consequently the estimation of cell means and covariances are not allowed. The problem of empty cells is more pronounced when a high number of categorical variables and clusters are involved, but the sample size is not sufficiently large (Franco, Crossa, Villasenor, Taba, and Eberhart, 1998). In the classification of genetic resources, Franco et al. (1998) proposed the modified location model which allows some of the *m* x *g* cells to remain empty. The modifications are that the means and covariances of the continuous variables depend only on the *k*-th cluster, instead of on the specific *ks*-th cell. The model assumes that the mean vectors and the covariance matrices are equal for all the multinomial cells within each cluster. Therefore the p.d.f becomes

$$f(\mathbf{x}_{si}, w_s; \mathbf{\Theta}) = \sum_{k=1}^{g} \alpha_k p_{ks} \phi(\mathbf{x}_{si}; \mu_{k,} \Sigma_k)$$

In term of parameters estimation, Franco et al. (1998) incorporated the technique suggested by Ward (1963) in the modified location model (MLM) and proposed a two-stage strategy, Ward-MLM, as an alternative to the different random starting points used by Lawrence and Krzanowski (1996). The basic notion of Ward-MLM is that the initial groups formed by the Ward method based on the objective function to minimize the sum of square within groups are used as the starting values for the EM algorithm as in equations (5.3a – e). The MLEs of α_k and p_{ks} for the modified location model are same as the conditional Gaussian model. However, the estimate of the cluster mean is the weighted MLEs of the conditional Gaussian model means, $\hat{\mu}_{ks}$, given by

$$\hat{\mu}_k = \sum_{s=1}^{m}\sum_{i=1}^{n_s}\frac{\hat{z}_{ksi}\mathrm{x}_{si}}{n\hat{\alpha}_k}\sum_{s=1}^{m}\hat{p}_{ks}\hat{\mu}_{ks} \ , \tag{5.4a}$$

and the estimate of the homogeneous covariance becomes

$$\hat{\Sigma} = \sum_{k=1}^{g}\sum_{s=1}^{m}\sum_{i=1}^{n_s}\frac{\hat{z}_{ksi}(\mathbf{x}_{si}-\hat{\mu}_k)(\mathbf{x}_{si}-\hat{\mu}_k)'}{n} \ . \tag{5.4b}$$

The probability of membership for each observation belonging to the *k*-th cluster is estimated as

$$z_{ksi} = \frac{\alpha_k p_{ks}\exp\left\{-\frac{1}{2}(x_{si}-\mu_k)'\Sigma^{-1}(x_{si}-\mu_k)\right\}}{\sum_{k=1}^{g}\alpha_k p_{ks}\exp\left\{-\frac{1}{2}(x_{si}-\mu_k)'\Sigma^{-1}(x_{si}-\mu_k)\right\}} \tag{5.4c}$$

Franco, Crossa, Taba, and Eberhart (2002) modified the modified location model by assuming heterogeneity of covariance matrices across clusters. The MLE of the covariance matrix of the *k*-th cluster is

$$\hat{\Sigma}_k = \sum_{s=1}^{m}\sum_{i=1}^{n_s}\frac{\hat{z}_{ksi}(\mathbf{x}_{si}-\hat{\mu}_k)(\mathbf{x}_{si}-\hat{\mu}_k)'}{n\hat{\alpha}_k} \ . \tag{5.5a}$$

The assignment of the observations to the *k*-th cluster assuming heterogeneity of covariances is done based on maximized a posterior probability given by

$$z_{ksi} = \frac{\alpha_k p_{ks}\exp\left\{-\frac{1}{2}(x_{si}-\mu_k)'\Sigma_k^{-1}(x_{si}-\mu_k)\right\}}{\sum_{k=1}^{g}\alpha_k p_{ks}\exp\left\{-\frac{1}{2}(x_{si}-\mu_k)'\Sigma_k^{-1}(x_{si}-\mu_k)\right\}} \ . \tag{5.5b}$$

This thesis proposes a mixture model for mixed data which is termed as the Gaussian location model (GLM) to solve the problems of indeterminacy of cluster label at each location and allow some of the *m* x *g* cells to remain empty. In the

GLM, a joint probability distribution of the mixed variables is proposed. The details of the model will be discussed in the next section.

5.2 Gaussian Location Model (GLM)

The Gaussian location model adopts the justification from the modified location model that the maximization process searches for homogeneous groups around the cluster mean and not around the cell mean helps to solve the empty cells problem. The proposed Gaussian location model introduces a joint probability distribution for mixed variables that produces a likelihood function in which each observation is compared with the cluster mean not with the cell mean, and at the same time obtains identifiable estimates of the model parameters. The proposed joint probability distribution for mixed variables is expressed as the conditional distribution of the categorical variables given the continuous variables, times the marginal distribution of the continuous variable. Actually, the idea of this form of joint probability distribution has been briefly raised by Cox (1972). He suggested that the joint distribution of a mixture of binary and continuous variables could be written as a logistic conditional distribution of the binary variables for given values of the continuous variables, times a marginal multivariate normal distribution for the continuous variables. However, this idea appears not to have been pursued any further in the analysis of mixed data sets. This thesis adopts Cox's idea of joint probability distribution for mixed variables and incorporates it in the mixture model for mixed data.

The proposed marginal distribution of the joint distribution is fitted from the Gaussian mixture distribution of the continuous variables, and the conditional distribution of the categorical variables is a multinomial distribution of the collapsed categorical variables given the Gaussian mixture distribution of the continuous variables. The resulting means and covariances from each cluster are equal for all the multinomial cells within the cluster. There are two main advantages of such a joint distribution: 1) like in the modified location model, the means and covariances of the continuous variables depend only on the cluster, and do not depend on the multinomial cell in which they appear, thus the assignment of observations into empty cells is allowed, and 2) in the finite mixture model for continuous variables, permutation of cluster labels corresponding to a simple relabeling of indexes, and the representation of different cluster labels are considered equivalent, thus the issue of identifiability is only up to relabeling of indexes and does not concern the cluster structure as in the conditional Gaussian model.

Let $\mathbf{y}_i = (\mathbf{u}_i, \mathbf{x}_i)$, be the mixed variables of q categorical variables, $\mathbf{u}_i = (u_{1i}, \ldots, u_{qi})$, and p continuous variables, $\mathbf{x}_i = (x_{1i}, \ldots, x_{pi})$, measured on the i-th observation of n units of a data set. Suppose that the q categorical variables as in the conditional Gaussian model can be uniquely transformed to a single multinomial random variable W with m cells, w_s (s = 1, …, m), where $m = \prod_{j=1}^{q} c_j$ is the number of distinct combination (location) of the q categorical variables. The data set is first clustered using only the continuous variables by fitting a G-component Gaussian mixture model on it as described in section 1.1.1. The clustering framework of Banfield et al. (1993), which allows features of cluster such as orientation, size and shape to vary across clusters, is applied here. The MLE of the model parameters, $\hat{\pi}_k$, $\hat{\mu}_k$, $\hat{\Sigma}_k$, and n_k (k = 1,…, G) are obtained using EM algorithm as in equations (1.9a–d). The given data set according to cluster is denoted by $\mathbf{w} = (w_{11}, \ldots, w_{1n_1}, w_{21}, \ldots, w_{2n_2}, \ldots, w_{G1}, \ldots, w_{Gn_k})'$, where w_{ki} is the location for the *i*-th out of the n_k observation in cluster *k* (*k* = 1,…, *G*; $\sum_{k=1}^{G} n_k = n$). The location of the observation is assumed multinomial distributed, given the observation is from cluster *k*. The probability of an observation in cluster *k* (*k* = 1, …, *G*) falls in cell w_s (*s* = 1, …, *m*) of the multinomial variable is

$$\hat{p}_{ks} = \sum_{i=1}^{n_k} \frac{\lambda_s \hat{z}_{ik}}{n_k} \tag{5.6}$$

where $\lambda_s = 1$ if $w_{ki} \in w_s$, and $\lambda_s = 0$ if $w_{ki} \notin w_s$.

The p.d.f. of each mixed data observation in the proposed mixture model is given by

$$f(\mathbf{x}_i, \mathbf{w} \mid \mathbf{\Psi}) = \sum_{k=1}^{G} \sum_{s=1}^{m} \pi_k p_{ks} \phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k), \tag{5.7}$$

and the posterior probability of membership is

$$\tau_{ksi} = \frac{\pi_k p_{ks} \phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k)}{\sum_{k=1}^{G} \pi_k \phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k)}. \tag{5.8}$$

In the Gaussian location model, the preliminary model fitting using only continuous variables provides the initial cluster structure and the determination of the number of clusters in the data, but overall, the information contained in both the continuous and categorical variables are used to assign the membership of the observations.

The estimations of the conditional Gaussian model and modified location model assume that the number of clusters is known. However, in practice, it is often unknown. The proposed Gaussian location model for mixed data has the advantage over the aforementioned models that the number of clusters can be estimated using well established model selection criteria like BIC.

5.3 Simulation for GLM

The performance of the proposed Gaussian location model is compared to the conditional Gaussian model (Lawrence & Krazanowski, 1996), the restricted location mixture model (Willse & Biok, 1999), and the modified location model (Franco et al., 2002).

Simulation studies 1 and 2 intend to examine the issue of identifiability of the proposed model, the conditional Gaussian model and restricted location mixture model by comparing the misclassification rates. For the restricted location mixture model, only the parallel structure in the conditional mean which assumes that the difference between conditional means for any two clusters is the same at all locations, was studied as it is more comparable with the proposed model. The simulated data in both of the simulations assume that the multinomial variable W is associated with the continuous variables. The generated data assumes homogeneity of covariances across clusters.

In simulation study 3, the evaluation focused on the ability of the proposed model compared to the modified location model in recovering the clusters structure in the presence of empty cells in the data set by clustering observations around the cluster means instead of the cell means. The modified location model considered in this simulation study is the Ward-MLM strategy which assumes that the covariance matrices are heterogeneous as in Franco et al. (2002). The simulated data assumes independence of the multinomial variable and the vector of continuous variables.

5.3.1 Simulation Study 1

The simulation examples of Lawrence and Krzanowski (1996) and Willse and Biok (1999) were used to compare the performance of the proposed mixture model for mixed data. Lawrence and Krzanowski (1996) conducted a simulation study to evaluate the ability of the location model to recover group structure and to classify observations. For each replication 20 observations were generated from each of two 4-variate normal populations, one with mean (0,0,1,1) and the other with mean (0,0,6,6), and the populations have common covariance matrix

$$\Sigma = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 3 \end{pmatrix}.$$

The first two variables for each observation were dichotomised by thresholding at 0. Let the generated value of the p-th continuous variable be x_p, if $x_p < 0$, then a binary value $y_p = 0$ is set, and if $x_p \geq 0$, $y_p = 1$, for p = 1, 2, in both populations. The whole process was replicated 50 times. Willse and Biok (1999) repeated the simulated data 50 times for the restricted location mixture model. In their simulation study, the EM algorithm was applied with randomly selected initial values and with initial values determined by applying k-means to the continuous variables. Both of the simulations by Lawrence et al. (1996) and Willse et al. (1999) assumed that the number of clusters was known where g = 2. This thesis also conducted 50 replications of the above simulated data for the proposed model, and used BIC to choose the number of clusters.

Table 5.1. Misclassification rates for simulation study 1 using location model, restricted location model and the proposed Gaussian location model.

Method	Misclassification (%)	
	mean	s.e.
Conditional Gaussian model (Lawrence et al., 1996)	31.4	1.95
Restricted location mixture model (Willse et al., 1999)	4.5	See text
Proposed GLM	2.98	0.49

Table 5.2. Average estimates and standard errors of the parameters means and covariances for the two continuous variables for simulation study 1 using location model, restricted location model, and the proposed Gaussian location model.

Method	Mean vectors		Covariance matrix	
	Average estimate	s.e.	Average estimate	s.e.
Conditional Gaussian model (Lawrence et al., 1996)	(2.52, 2.52), (4.47,4.47)	≈ 0.1	$\begin{pmatrix} 1.753 & 0.883 \\ 0.883 & 2.320 \end{pmatrix}$	$\begin{pmatrix} 0.185 & 0.173 \\ 0.173 & 0.119 \end{pmatrix}$
Restricted location mixture model (Willse et al., 1999)	(1.00, 0.98), (5.96, 5.96)	≈ 0.10	$\begin{pmatrix} 2.14 & 1.08 \\ 1.08 & 2.86 \end{pmatrix}$	≈ 0.05 for all entries
Proposed model	(0.86, 0.91), (6.08, 6.11)	≈ 0.06	$\begin{pmatrix} 1.979 & 0.420 \\ 0.420 & 2.349 \end{pmatrix}$	$\begin{pmatrix} 0.084 & 0.081 \\ 0.081 & 0.077 \end{pmatrix}$

The misclassification rates for the simulated data are compared in Table 5.1. For the proposed Gaussian location model, 7 out of 50 replications of the simulated data did not choose the number of clusters to be 2. These replications were omitted in the report. For the restricted location mixture model, one of the simulated data set contains no observations from one of the location, and it was not taken in account in the summary statistics in Table 5.1. It can be seen that the proposed model performed better than the others and, in particular, it is far better than the conditional Gaussian model. Two of the replications in the proposed model contained empty cell. However they did not affect the estimation of parameters, and the replications give an average of 1.25% misclassification rate. According to Willse et al. (1999), the true model for this set of data is the underlying variable mixture model of Everitt (1988) which assumes that the binary variables are obtained by dichotomising underlying normal variables, and the misclassification rate estimated by the underlying variable mixture model is 1.1%. It shows that the misclassification rate of the proposed model is closer to true model as compared to other models. In Willse et al. (1999), the standard error of the misclassification rate was not reported, however from other statistics for the misclassification rate: median 2.5%, minimum 0%, maximum 22.5%, it shows that the restricted location mixture model has high variability in term of classification accuracy.

The average estimates of the parameters are shown in Table 5.2. The two population mean vectors for the continuous variables are (1,1) and (6,6). The conditional Gaussian model shows shrinkage of means towards the centre of both variates. Both restricted location mixture model and the proposed model recovered the parameters of means well. The true covariance matrix according to Everitt's (1988) model as cited in Willse et al. (1999) is

$$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 2.5 \end{pmatrix}.$$

It can be seen that the average estimate of covariance in the proposed model is the closest to the true value.

Even when there is some degree of dependency between the continuous variables and the multinomial variable obtained from the dichotomised variables, the proposed model performed well.

5.3.2 Simulation Study 2

A simulated data consists of less well separated groups studied by Willse et al. (1999) was used to compare the performance of the proposed model and the restricted location mixture model. The common covariance matrix of the populations, and the threshold to dichotomize the first two variables are same as simulation study 1. The two populations in simulation study 2 have closer means, one with mean (1,0,5,5) and another one with mean (0,1,2,2). For each replication, 100 observations were generated for each of the two populations.

The misclassification rates for simulation study 2 for the two models are shown in Table 5.3. For the proposed model, BIC did not suggest the correct number of clusters for 38 out of 50 replications. However, in the replications where the numbers of clusters were correctly determined, the proposed model outperformed the restricted location mixture model in term of classification accuracy. The respective average misclassification rate is 13.72% which is much lower than 21.45% for the restricted location mixture model as reported in Willse et al. (1999). Furthermore, from the other statistics of the misclassification rate in Willse et al. (1999) like median 18.75%, minimum 8.0%, and maximum 41%, the restricted location mixture model showed much broader variation between its classification rate than the one obtained by the proposed model. However, the obtained misclassification rate by the proposed model is still a bit higher that the true misclassification rate under Everitt's (1988) model which is only 7.2%.

Table 5.3. Misclassification rate for simulation study 2 using restricted location model and the proposed Gaussian location model.

Method	Misclassification (%)	
	mean	s.e.
Restricted location mixture model (Willse et al., 1999)	21.45	See text
Proposed model	13.46	0.82

5.3.3 Simulation Study 3

In simulation study 3, a simulated data from Franco et al. (2002) was used to compare the performance of the proposed model and the modified location model (MLM). Franco et al. (2002) generated four multivariate normal clusters with heterogeneous covariance matrices using the population parameters from Taba, Diaz, Franco and Crossa (1998) for two continuous variables: days to silk, V_1, and plant height, V_2, given as in Table 5.4, where μ_{kp} and σ_{kp}^2 are the means and variances respectively associated with the *k*-th cluster of size n_k for variable *p* (*p* = 1, 2), σ_{k12} is the covariance between the bivariate. Assignment of the number of observations for each level of the multinomial variable, *W*, i.e.1, 2 and 3 is according to the distribution in Table 5.4. It can be seen that the cells in level 3 of cluster 1 and 4 and level 1 of cluster 2 and 3 are empty. Franco et al. (2002) run the simulated data for once. Fifty replications were carried out for the proposed model.

Table 5.4. The means, covariances of two continuous variables, number of observations for each level of the multinomial variables, for four groups from Franco et al. (2002).

k	μ_{k1}	μ_{k2}	σ^2_{k1}	σ^2_{k2}	σ_{k12}	Levels of *W*			n_k
						1	2	3	
1	60	190	10	350	35	20	5	0	25
2	70	180	22	270	30	0	5	20	25
3	85	225	32	400	4	0	5	20	25
4	100	240	10	350	58	20	5	0	25

Table 5.5. Means, variances, covariances, and number of observations for each level of the multinomial variable for four clusters in simulation study 3 obtained using the modified location model and the proposed model.

Method	*k*	$\hat{\mu}_{k1}$	$\hat{\mu}_{k2}$	$\hat{\sigma}^2_{k1}$	$\hat{\sigma}^2_{k2}$	$\hat{\sigma}_{k12}$	Levels of *W*			n_k
							1	2	3	
Modified location model (Franco et al., 2002)	1	59.8	188.1	7.5	423.4	39.95	20	5	0	25
	2	79.1	219.0	106.8	923.0	290.42	0	**10**	**0**	10
	3	76.8	196.3	70.9	784.7	185.87	0	**0**	**40**	40
	4	99.8	238.1	7.5	305.3	46.85	20	5	0	25
Proposed model#	1	61.1	193.5	9.6	310.7	23.23	20	5	0	25
	2	72.1	188.9	41.4	270.4	42.13	0	**7**	**24**	31
	3	86.8	226.5	29.1	272.1	-0.73	0	**3**	**15**	18
	4	100.0	240.4	9.6	340.2	56.05	20	5	**1**	26

Note #: The estimates of parameters are average estimates of the replications. The distribution of observations according to levels of *W* is based on one of a replication that shows misclassification rate of 8% (mode).

Table 5.5 presents the characteristics of the cluster structure obtained using the modified location model and the proposed model for simulation study 3. The average estimates of parameters in the proposed model are much closer to the population parameters compared to the modified location model. The average misclassified rate of the proposed model is 11.13% with standard error 2.5. This is lower than the misclassified rate of 20% for the modified location model as reported in Franco et al. (2002). From Table 5.5, it can be seen that the proposed model also recovered the clusters structure better according to the level of the multinomial variable in the presence of empty cells. The modified location model successfully recovered the observations points in cluster 1 and 4 even when the clusters structure in terms of means and covariances deviate from the true values. However, all the 5 observations from level 2 of cluster 2 were misclassified in level 2 of cluster 3, and all the 20 observations from level 3 of cluster 2 were misclassified in level 3 of cluster 3. The modified location model is likely to produce more empty cells.

5.4 Scalable Clustering Algorithm for Mixed Data

Existing mixture models that can handle mixed types of attributes are not efficient when clustering large data sets. In this thesis, a scalable clustering algorithm based on mixture model that can handle very large mixed data sets, FlexClustMix, is proposed. FlexClustMix adapts the algorithm from FlexClust which has been explained in detail in Chapter 4. There are two main modifications with respect to: 1) the data compression method and the resulting prototype system, and 2) updating of parameters.

In FlexClustMix, the observation points from the random sample of mixed data in the memory buffer are fitted by the proposed Gaussian location model as described in section 5.2. The identified dense regions are compressed according to the continuous variables space and represented by a prototype system consists of MLEs of the Gaussian location model

$$\hat{\mathbf{\Psi}} = (\hat{\alpha}_1,...,\hat{\alpha}_g,\hat{\mu}_1,...,\hat{\mu}_g,\hat{\Sigma}_1,...,\hat{\Sigma}_g,\hat{p}_{11},...,\hat{p}_{1m},...,\hat{p}_{g1},...,\hat{p}_{gm})\,. \tag{5.8}$$

The prototype system from (5.8) is decomposed to its mixture components

$$\hat{\Psi}_k = (\hat{\mu}_k, \hat{\Sigma}_k, \hat{p}_{ks}, n_k), \qquad k = 1, \ldots, g; s = 1, \ldots, m$$

where $n_k = \hat{\alpha}_k n$ is the *k*-th cluster size.

The compressed clusters in Gaussian location model are assumed to be the same as the clusters in Gaussian mixture model. Thus, in FlexClustMix, the identification of the nearest neighbour cluster and the determination of merge of nearest neighbour pair are done in the same way as FlexClust.

Model updates for FlexClustMix are carried out incrementally based on the suggestion from the MBF criteria. The update of parameters $\hat{\mu}_k$, $\hat{\Sigma}_k$ n_k are similar to the FlexClust algorithm as described in section 4.3.2.3. For the parameter $\hat{p}_{ks}$, consider $\hat{p}_{1s}$ and $\hat{p}_{2s}$ which are the parameters of the nearest neighbour pair, and $\hat{p}_{Ms}$ is the corresponding parameter of the merged cluster. If the MBF suggests merging of the nearest neighbour pair, the parameter of the probability of an observation in cluster *k* (*k* = 1, …, *G*) falls in cell w_s (*s* = 1, …, *m*) of the multinomial variable in the current model becomes

$$\hat{p}_{Ms} = \frac{\hat{p}_{1s} n_1 + \hat{p}_{2s} n_2}{n_1 + n_2}, \qquad \text{for } s = 1, \ldots, m.$$

On the other hand, if the MBF criterion suggests adding a new cluster, $\hat{p}_{2s}$ is the corresponding parameter for the new cluster.

5.5 Simulation for FlexClustMix

In this section, the effectiveness of the compression method and model updates using the proposed Gaussian location model in FlexClustMix was first evaluated using a small scale data. Consider the number of compressed samples is only two and the incorporation of the later compressed information into the current model was carried out once. The basic idea is similar to the bi-Gaussian mixture model in Chapter 3. Thus the resulting model is called the bi-Gaussian location model, since the Gaussian location model is used for data compression instead of Gaussian mixture model. The performance of the proposed scalable FlexClustMix algorithm is then tested on large simulated data.

Simulation for Bi-Gaussian Location Model

The simulated data in Lawrence et al. (1996) is revisited to assess the effectiveness of the Gaussian location model in preserving the clusters structure during data compression, and the accuracy of parameters update of the current model through the bi-Gaussian location model. Three experiments were designed to assess the accuracy of the estimates of parameters. For each experiment, 50 replications are considered. First, the simulated data were generated twice the original size, and divided into two samples: sample 1, S_1, and sample 2, S_2, according to different sample size ratios using separate sampling scheme so that the mixture proportions in both samples are not known. Secondly, the simulated data in Lawrence et al. (1996) is used as sample 1, and the data for sample 2 are generated from the same parameters of means and covariances as in sample 1 but the mixture proportions and sample sizes are different. Lastly, the experiment studied the ability of the proposed model selection criterion, MBF, in identifying the change of clusters structure, and the accuracy of the update of new cluster in the current model through the bi-Gaussian location model. For this purpose, sample 2 is added with a new cluster consists of 35 observations points generated from a 4-variate normal population mean (0,0,3,3) and have common covariance matrix as the other two clusters in sample 2, and the first two variables for each observation were dichotomised by thresholding at 0 according to Lawrence et al. (1996). The performance of the bi-

Gaussian location model is compared to the Gaussian location model obtained using the combined data of sample 1 and sample 2.

Table 5.6. Average estimates of means and covariances and misclassification for the bi-Gaussian location model and the Gaussian location model on the simulated data. The bi-Gaussian location model fitted on the simulated data that divided into two samples according to different ratios. The Gaussian location model fitted on the combined data.

Bi-Gaussian location model from 2 samples				Gaussian location model from combined data		
Ratio S_1: S_2	Mean vectors	Covariance matrix	m.r. (%)	Mean vectors	Covariance matrix	m.r. (%)
1:1	(0.98, 0.96), (6.07, 6.00) s.e. = 0.05	$\begin{pmatrix} 1.844 & 0.721 \\ 0.721 & 2.817 \end{pmatrix}$ s.e. = 0.09	2.58 s.e. = 0.3	(1.04, 0.98), (6.07, 6.04) s.e. = 0.04	$\begin{pmatrix} 1.914 & 0.966 \\ 0.966 & 2.833 \end{pmatrix}$ s.e. = 0.06	2.69 s.e. = 0.3
3:5	(1.03, 0.97), (6.06, 6.00) s.e.=0.06	$\begin{pmatrix} 1.782 & 0.510 \\ 0.510 & 2.677 \end{pmatrix}$ s.e. = 0.10	1.85 s.e. = 0.3	(1.04, 0.98), (6.07, 6.04) s.e. = 0.04	$\begin{pmatrix} 1.914 & 0.966 \\ 0.966 & 2.833 \end{pmatrix}$ s.e. = 0.06	2.69 s.e. = 0.3
5:3	(0.99, 0.96), (6.04, 6.06) s.e. = 0.05	$\begin{pmatrix} 1.774 & 0.737 \\ 0.737 & 2.851 \end{pmatrix}$ s.e. = 0.09	2.45 s.e. = 0.4	(1.04, 0.98), (6.07, 6.04) s.e. = 0.04	$\begin{pmatrix} 1.914 & 0.966 \\ 0.966 & 2.833 \end{pmatrix}$ s.e. = 0.06	2.69 s.e. = 0.3

Note: S_i = sample i, i = 1, 2; m.r. = misclassification rate

Table 5.7. Average estimates of means and covariances and misclassification for the bi-Gaussian location model and the Gaussian location model on the simulated data. Data for sample 2 in the bi-Gaussian location model are generated from different mixture proportions and with new cluster added. The Gaussian location model fitted on the combined data.

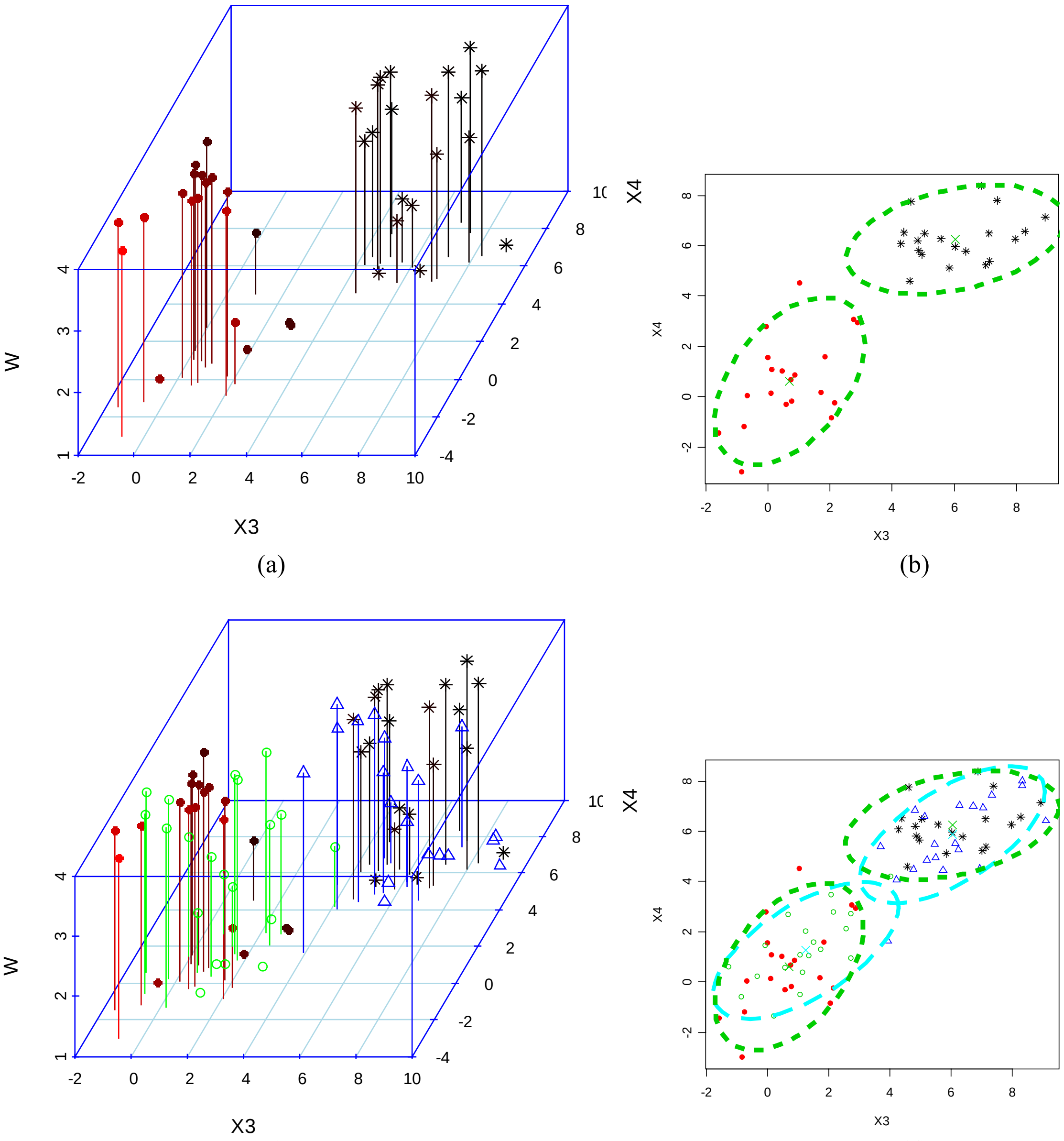
Sample 2		Bi-Gaussian location model from 2 samples			Gaussian location model from combined data		
Mixture proportions (λ_1, λ_2)	size	Mean vectors	Covariance matrix	m.r. (%)	Mean vectors	Covariance matrix	m.r. (%)
(0.5,0.5)	40	(0.98, 0.95), (6.02, 5.97) s.e. = 0.05	$\begin{pmatrix} 1.982 & 0.551 \\ 0.551 & 2.589 \end{pmatrix}$ s.e. = 0.09	2.86 s.e. = 0.3	(0.98, 0.95), (6.02, 5.98) s.e. = 0.05	$\begin{pmatrix} 1.990 & 0.666 \\ 0.666 & 2.572 \end{pmatrix}$ s.e. = 0.09	2.95 s.e. = 0.3
(1/3,2/3)	60	(0.93, 1.01), (6.04, 6.03) s.e.= 0.04	$\begin{pmatrix} 1.958 & 0.750 \\ 0.750 & 2.801 \end{pmatrix}$ s.e. = 0.07	2.74 s.e. = 0.2	(0.92, 1.00), (6.04, 6.02) s.e.= 0.04	$\begin{pmatrix} 1.908 & 0.892 \\ 0.892 & 2.825 \end{pmatrix}$ s.e. = 0.06	3.11 s.e. = 0.4
(0.25,0.4)+ new cluster	100	(0.94, 0.95), (6.00, 6.01), (3.03, 10.02) s.e. = 0.04	$\begin{pmatrix} 2.001 & 0.751 \\ 0.751 & 2.734 \end{pmatrix}$ s.e. = 0.07 For new cluster: $\begin{pmatrix} 1.995 & 0.932 \\ 0.932 & 2.927 \end{pmatrix}$ s.e. = 0.06	3.69 s.e. = 0.3	(0.94, 0.97), (5.93, 6.11), (3.09, 9.94) s.e. = 0.11	$\begin{pmatrix} 1.920 & 0.889 \\ 0.889 & 2.871 \end{pmatrix}$ s.e. = 0.04	3.73 s.e. = 0.3

Note: m.r. = misclassification rate

The results for misclassification rate and average estimate of parameters for models obtained through the bi-Gaussian location model and Gaussian location model on the simulated data are shown in Table 5.6 and Table 5.7. Overall, the model obtained by compressing sample data using the Gaussian location model and updating the model based on bi-Gaussian location model shows parameter estimates that are very close to the model fitted on the combined data using the Gaussian location model. For the misclassification rates, the models obtained from the bi-Gaussian location model according to different sample sizes ratios are 2.5%, 1.85% and 2.45%, which are slightly lower than the one obtained by the Gaussian location model, i.e. 2.69%.

From Table 5.6, it can be seen that even under separated sampling scheme, the estimates of parameters from the bi-Gaussian mixture model can be recovered close to the true population parameters values as given in simulation study 1 in section 5.3.1, regardless of the sample sizes ratios of sample 1 to sample 2.

Table 5.7 shows that the mixture proportions and size of sample 2 do not significantly affect the update of current Gaussian location model fitted on the basis of sample 1. The obtained estimates of parameters are very close to the ones obtained by fitting the combined data of sample 1 and sample 2. The differences of the misclassification rates are between the range of 0.09 (=2.95-2.86) and 0.37 (=3.11-2.74). Out of 50 replications, there is 1 replication with an empty cell in sample 1, and 2 replications with empty cells in sample 2. The presence of empty cells do not matter in the earlier compressed data or the later compressed data did not affect the estimates of parameters through the bi-Gaussian location model in these 3 replications and give an average of 2.1% misclassification rate. Figure 5.1 shows one of these examples. In Figure 5.1 (a) the 3D plot of sample 1 shows that the cell at level 3 of multinomial variable W for the cluster represented by red dots is empty. Compression around the cluster means as shown in Figure 5.1 (b) represents the prototype system conditional on the distribution of the continuous variables. When the compressed sample 2 was used to update the current model around the cluster means as shown in Figure 5.1 (d), all the cells conditional on the continuous variables were updated regardless it was empty or not, without affecting the existing number of clusters.



(a)

(b)

(c)

(d)

Figure 5.1. Plot of the simulated data for (a) sample 1 in 3D using the mixed data, (b) sample 1 using the continuous variables, (c) combined data of sample 1 and 2 in 3D using the mixed data, and (d) combined data of sample 1 and 2 using the continuous variables. (Red dot and black star are two clusters of sample 1; green circle and blue triangle are two clusters of sample 2).

When sample 2 with a new cluster was compressed and used to update the current Gaussian location model, the model selection model MBF, managed to detect the change in clusters structure. Furthermore, the estimate of parameters obtained from the update of current model through the bi-Gaussian location model is a good approximation to the model fitted on the combined data using Gaussian location model as shown in Table 5.7. The misclassification rates for both of the model are 3.69% and 3.73% respectively.

Simulated Large Data

The data set generated using the means and covariance matrix of five continuous variables (Days to anthesis (V_1), days to silk (V_2), plant height (V_3), ear height (V_4), and grain moisture (V_5)) from a maize evaluation trial from USA (Taba, Diaz, Franco, Crossa and Eberhart, 1999) considered by Franco and Crossa (2002) was applied in this section to generate a large data set consists of 1 million data points. The data set assumes a high degree of overlap on the multinomial variable, and also all the continuous variables with indistinct boundaries. The covariance matrix, *S*, and the corresponding correlation matrix, *R*, of the real data are given by

$$S = \begin{pmatrix} 95 & 102 & 439 & 371 & -13 \\ 102 & 112 & 479 & 402 & -14 \\ 439 & 479 & 3019 & 2255 & -158 \\ 371 & 402 & 2255 & 1849 & -127 \\ -13 & -14 & -158 & -127 & 52 \end{pmatrix}, R = \begin{pmatrix} 1 & 0.99 & 0.82 & 0.89 & -0.19 \\ 0.99 & 1 & 0.82 & 0.88 & -0.18 \\ 0.82 & 0.82 & 1 & 0.95 & -0.40 \\ 0.89 & 0.88 & 0.95 & 1 & -0.41 \\ -0.19 & -0.18 & -0.40 & -0.41 & 1 \end{pmatrix}.$$

In Franco and Crossa (2002), the values (m – 2s), (m + 2s), (m + 6s), (m + 10s), and (m + 14s), where m is the overall observed means and s is the standard deviation vector from the five variables, were assigned as the simulated cluster means, allowing a distance of 4s between means of neighbour clusters on each variable. Two of the continuous variables, V_2 and V_4, were used to generate two categorical variables. From the correlation matrix, it is known that there is high correlation between V_2 and V_1, and between V_4 and V_3 respectively. The first categorical variable is a binary variable that takes value 0 if $V_2 \leq \overline{V_2}$, where $\overline{V_2}$ is the mean of V_2, and takes value of 1 if $V_2 > \overline{V_2}$. The second categorical variable is a multi-state variable discretized from V_4. It takes values 1, 2, 3 and 4 for values of V_4 within (P_0, $P_{0.25}$), ($P_{0.25}$, $P_{0.5}$), ($P_{0.5}$, $P_{0.75}$), ($P_{0.75}$, P_1) respectively, where P_p is the p-th percentile of V_4. In this simulation, the sample size considered for filling in the memory buffer for the FlexClustMix algorithm is 1,000, and the value of m was set at (0,0,0,0,0). Only one replicate was simulated for this set of data.

The proposed FlexClustMix algorithm estimated the number of clusters correctly, and recovered the clusters structure very well with misclassification rate 0.015%. Result of this simulation shows that FlexClustMix is not only able to handle large data set, it is also robust under strong dependence between variable W and the continuous variable, and even overlap on the continuous and categorical variables.

5.6 Conclusion

The present conditional Gaussian model and the derivations of it suffer from estimation problem when applied in many practical situations because of the large number of parameters the models contain. This also causes the limitation of using model-based method for clustering very large set of mixed data. This thesis first proposes a modified mixture model for clustering mixed data, Gaussian location model, to reduce the number of parameters involved in parameters estimate during the EM algorithm, and at the same time solves problem of non-identifiability and empty cells as shown in simulation study 1, 2 and 3. This thesis also develops a scalable mixture model clustering algorithm for mixed data, FlexClustMix, based on the successes of the proposed Gaussian location model and the FlexClust algorithm. The algorithm was tested over dependent categorical and continuous variables, and change of clusters structure, through the proposed bi-Gaussian location model. The results obtained in term of accuracies of parameters estimation and recovery of clusters structure are highly encouraging. It is also remarkable to notice that the FlexClustMix algorithm able to handle very large mixed data efficiently and effectively as demonstrated in the simulation result.

6

Conclusion and Future Work

In this chapter, the contributions of the thesis are summarized. Then, the possible application of the proposed clustering algorithms is discussed. Finally, the limitations of the proposed clustering algorithm are discussed and some possible future directions to further enhance the work of this thesis are given.

6.1 Contributions

This thesis focuses on developing mixture model clustering algorithms for very large data sets that do not fit into the computer memory buffer. Two algorithms, FlexClust and FlexClustMix are proposed respectively for numerical data and mixed data. The basic notion of these two algorithms is to compress data incrementally according to the available memory buffer using the appropriate mixture model according to the type of data, and incorporating the compressed information into the current model with the flexibility of allowing changes in the clusters structure and the number of clusters.

Few new frameworks and models have been proposed to support the two algorithms mentioned above. First, a new semi-supervise learning framework for mixture model is proposed in Chapter 3. It considers updating trained Gaussian mixture model on the basis of unclassified data to have been drawn from or outside the underlying population. A model selection criterion, MBF, is proposed to determine whether the unclassified data has the same distribution as the model trained by the classified data for discovering new clusters. The update of trained model is carried out based on the proposed bi-Gaussian mixture model where only the MLEs of the parameters from the trained Gaussian mixture model and the Gaussian mixture model fitted on the unclassified data are involved. The results are compared to the Gaussian mixture model fitted on the combined data of training and unclassified data. It shows that both models give very close estimates of parameters and classification accuracies. This implies that the bi-Gaussian mixture model which updates current mixture model directly using summary statistics works effectively.

FlexClust extends the framework of bi-Gaussian mixture model to cluster very large set of continuous data in Chapter 4. The algorithm adapts the incremental compression procedure which maintains only the summarized information in the memory and purges the data points to free some memory for filling new data points to the memory buffer, and this makes it scalable to very large datasets. The incremental compression procedure is closely related to the current fit of the clustering model and provides usable model at any time. The performance of FlexClust is compared to some sampling-based algorithms such as Strategy III and CLARA, and compression-based algorithms such as sufficient EM and SPSS TwoStep as shown in Table 6.1. The results show that FlexClust outperforms these algorithms. The problems of non-representative sample and missing out of small clusters in the sampling-based algorithms, and the problem of scalability in the one time compression-based algorithms are overcame in FlexClust. Furthermore, the proposed FlexClust employs mixture modelling for data compression has reduced the loss of information caused by ineffective prototype system, and has obtained more consistent and accurate estimate of model parameters.

Table 6.1. Comparison of results for the proposed FlexClust and the other compression-based and sampling-based methods.

Algorithm	Data set	Size	Number of clusters	Dimension	Mean misclassification (%)	Identify correct clusters (%)	Other assessment of performance	Execution time
FlexClust compare to one-time compression methods:								
Sufficient EM	1	15,000	7	2	2.37% higher	50% better	-	1.4 times slower
	2	10,000	3	4	0.15% lower	Same	-	3.7 times slower
	3	20,000	3	13	50.88% lower	65% better	-	1.6 times faster
	4	87,472	-	3	-	-	Image equally good	5.7 times slower
	5	581,012	-	5	-	-	Loglikelihood 22,720 higher	5.1 times faster
SPSS TwoStep	1	15,000	7	2	10.50% lower	75% better	-	-
	2	10,000	3	4	28.13% lower	100% better	-	-
	3	20,000	3	13	0.11% lower	Same	-	-
	4	87,472	-	3	-	-	-	-
	5	581,012	-	5	-	-	-	-
FlexClust compare to random sampling methods:								
Strategy III	1	15,000	7	2	0.08% higher	75% better	-	14.3 times slower
	2	10,000	3	4	3.87% lower	85% better	-	22.5 times slower
	3	20,000	3	13	1.97% lower	95% better	-	3.9 times slower
	4	87,472	-	3	-	-	Recover small clusters	13.2 times slower
	5	581,012	-	5	-	-	Loglikelihood 66,250 higher	1.2 times slower
CLARA	1	15,000	7	2	12.30% lower	75% better	-	-
	2	10,000	3	4	28.14% lower	100% better	-	-
	3	20,000	3	13	32.28% lower	100% better	-	-
	4	87,472	-	3	-	-	-	-
	5	581,012	-	5	-	-	-	-

A parametric model for clustering mixed data, the Gaussian location model, is proposed in Chapter 5. The main advantages of the Gaussian location model are that it reduces the number of parameters involved in parameters estimate during the EM algorithm, and at the same time solves the problem of non-identifiability and empty cells which suffered in the conditional Gaussian model and its derivations. Simulation results as shown in Table 6.2 show that the Gaussian location model superior to the conditional Gaussian model, restricted location mixture model and modified location model by sufficiently reducing the loss of clustering accuracy.

Table 6.2. Comparison of results for the proposed Gaussian location model and other parametric models for clustering mixed data.

Model	Data set	Size	Number of clusters	Dimension of continuous variables	Level of multinomial	Mean misclassification (%)	Recovery of empty cells (%)
Gaussian location model compare to:							
Conditional Gaussian model	1	40	2 well separated	2	4	28.42% lower	-
Restricted location model	1	40	2 well separated	2	4	1.52% lower	-
	2	200	2 less well separated	2	4	8.00% lower	-
Modified location model	3		4 with 4 empty cells	2	3	9.07% lower	36% better

By adapting the framework of bi-Gaussian mixture model, a similar model, bi-Gaussian location model, is proposed to update the current model using a later compressed mixed data sample. The parameter estimates and classification accuracy of the bi-Gaussian location model are very close to the model fitted on the combined data using the Gaussian location model.

FlexClustMix incorporates the proposed Gaussian location model into the FlexClust algorithm to develop a scalable clustering algorithm for very large set of mixed data. Simulation result shows that the FlexClustMix algorithm is able to handle very large mixed data efficiently and effectively even under strong dependence between the multinomial variable and the continuous variables, and even overlap on the continuous and categorical variables.

As the proposed algorithms apply data compression for scalability, the challenge of avoiding loss of information becomes another focus of this thesis. This thesis proposes to use mixture models to tackle the problem of loss of information. The results on the simulated and real data show that working on summary statistics of subclusters compressed by mixture models preserves the cluster structure better compared to the other methods.

6.2 Application

Although the scalable clustering algorithms presented in this thesis are studied in the context of data size that does not fit in the memory buffer, they can also be used for clustering open data stream. In a streaming environment, the original data is discarded and only the summary is kept and the goal is to produce high quality summary of the data. Aggarwal, Wang and Yu (2003) and Barbara (2002) had set two main requirements for effective and efficient clustering of stream: 1) one pass over the data during on-line where the data points must be read in an incremental fashion and discarded in favour of summary, and 2) maintain compact and representative summary of the data during off-line. For open stream, the understanding of the underlying clustering structure of the data at different time may be required from this summary, and this structure may change as more data is processed. The proposed FlexClust and FlexClustMix algorithms satisfy the above requirements and therefore can be applied for clustering open stream.

6.3 Limitation and Future work

The data sets considered in this thesis are free from noise. The proposed scalable Gaussian mixture based clustering algorithms are not robust to noisy data. Peel and MacLachlan (2000) found that if a set of *G*-component normal mixture data in the presence of uniform background noise is fitted by a Gaussian mixture, the model selection criteria such as BIC and AIC, and bootstrapping of $-2\log\lambda$ attempt to model the background noise with an additional component. However, the estimate of model parameters is affected by the presence of noise. The contributions in this thesis can be improved along the way to establish robust scalable clustering algorithm.

One way to handle the presence of noise or atypical observations when fitting Gaussian mixture model is to include an additional component to describe the noise. Fraley and Raftery (1998) modeled the noise component as a constant-rate Poison process. The initial noise estimate is obtained using methods for denoising which includes Voronoi method (Allard & Fraley, 1997) and nearest neighbour method (Byers & Raftery, 1998). However, this method is sensitive to the value of the hypervolume of the data region. Another way of modelling the noise component is by using the uniform distribution. However, this model cannot work well in the situation when the noise is not uniform.

For future work, the initial idea of this thesis is to use mixture of *t* distributions to develop robust scalable clustering algorithm. The *t* distribution has a wider tail than the Gaussian distribution, and it is usually adopted alternatively as a standard choice for robustness. The mixture of *t* distributions provides a framework to assess the robustness of inclusion of atypical observations in the fitting of mixture model through the estimation of the degree of freedom of the *t* components p.d.f.

The data sets used in this thesis are of reasonable dimension. However, applications in data mining often lead to very high dimensional data. Clustering such data is challenging. Many recently developed clustering algorithms have attempted to address either handling data sets with very large number of observations or very high dimension. Successful scalable clustering algorithms must avoid the curse of dimensionality and at the same time overcome the scalability problems associated with very large data. The knowledge to scale up model-based clustering algorithm for high dimensional data sets is an area of interest for future research.

References

Aggarwal, C.C., Han, J, Wang, J. & Yu, P.S. (2003). A framework for clustering evolving data streams. In J.C. Freytag, P.C. Lockemann, S. Abiteboul, M.J. Carey, P.G. Selinger & A. Heuer (Eds.), *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB)* (pp. 81-92). Berlin: Morgan Kaufmann.

Agrawal, R., Gehrke, J., Gunopulos, D. & Raghavan P. (2005). Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery, 11* (1), 5-33.

Aitken, M. & Aitkin, I. (1996). A hybrid EM/Gauss-Newton algorithm for maximum likelihood in mixture distributions. *Statistics and Computing, 6*, 127-130.

Allard, D. & Fraley, C. (1997). Nonparametric maximum likelihood estimation of features in spatial point processes using Voronoi tessellation. *Journal of the American Statistical Association, 92*, 1485-1493.

Ankerst, M., Breunig, M.M., Kriegel, H.P. & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. In A. Delis, C. Faloutsos & S. Ghandeharizadeh (Eds.), *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (pp. 49-60). New York: ACM.

Asuncion, A. & Newman, D.J. (2007). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. Retrieved January 5, 2008 from http://www.ics.uci.edu/~mlearn/MLRepository.html

Banfield, J.D. & Raftery, A.E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics, 49*, 803-821.

Barbara, D. (2002). Requirements for clustering data streams. *SIGKDD Explorations, 3*(2), 23-27.

Bellman, R.E. (1961). *Adaptive Control Processes*. Princeton, NJ: Princeton University Press.

Beyer, K.S., Goldstein, J., Ramakrishnan, R. & Shaft, U. (1999). When is "nearest neighbor" meaningful? In C. Beeri & P. Buneman (Eds.), *Proceeding of the 7th International Conference on Database Theory* (pp.217-235). London: Springer-Verlag.

Bock, H.H. (1996). Probabilistic Models in Cluster Analysis. *Computational Statistics and Data Analysis*, *23*, 5–28.

Bradley, P.S., Fayyad, U. & Reina, C. (1998a). Scaling clustering algorithms to large databases. In R. Agrawal & P. Stolorz (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)* (pp. 9-15). Menlo Park, CA: AAAI Press.

Bradley, P.S., Fayyad, U. & Reina, C. (1998b). *Scaling EM (Expectation-Maximization) to large databases* (No. Technical Report MSR-TR-98-35). Redmond, WA: Microsoft Corporation

Bradley, P.S, Fayyad, U. & Reina, C.R. (2000). Clustering very large databases using EM mixture models. In *Proceedings of 15th International Conference on Pattern Recognition* (Vol. 2, pp. 76-80). Los Alamitos, CA: IEEE Computer Society Press.

Breunig, M.M., Kriegel, H., Kroger, P. & Sander, J. (2001). Data Bubbles: Quality preserving performance boosting for hierarchical clustering. In W.G. Aref (Ed.), *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data* (pp. 79-90).

New York: ACM Press.

Byers, S.D. & Raftery, A.E. (1998). Nearest neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association, 93,* 577-584.

Celeux, G. & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*(28), 781-793.

Celeux, G., Hurn, M. & Robert, C.P. (2000). Computational and inferential difficulties with mixture posterior distributions. *Journal of American Statistical Association, 95*, 957-970.

Cheeseman, P. & Stutz, J. (1995). Bayesian Classification (AutoClass): Theory and Results. In D.J. Jackson, & E.F. Borgotta (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 215-246). Beverly Hill: Sage Publications.

Chiu, T., Fang, D., Chen, J., Wang, Y. & Jeris, C. (2001). A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 263-268). New York: ACM Press.

Chou, W. & Reichl, W. (1999). Decision tree state tying based on penalized Bayesian information criterion. In *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 345-348). Washington, DC: IEEE Computer Society.

Clogg, C.C. (1988). Latent class models for measuring. In R. Langeheine & J. Rost (Eds.), *Latent Trait and Latent Class Models* (pp. 173-205). New York: Plenum Press.

Cox, D. R. (1972). The analysis of multivariate binary data. *Applied Statistics, 21*, 113-120.

Cozman, F. & Cohen, I. (2002). Unlabeled data can degrade classification performance of generative classifiers. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference* (pp. 327–331). Menlo Park, CA: AAAI Press.

Cozman, F.G., Cohen, I. & Cirelo, M.C. (2003). Semi-Supervised learning of mixture models. In T. Fawcett & N. Mishra (Eds.), *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)* (pp. 99-106). Washingtion, DC: AAAI Press.

Cutting, D.R., Karger, D.R., Pedersen, J.O. & Tukey, J.W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 318-329). New York: ACM Press.

Dasgupta, A. & Raftery, A.E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association, 93*(441), 294-3021.

Davidson, I. & Satyanarayana, A. (2003, November 19). *Speed up k-means clustering by bootstrap averaging.* Paper presented at the Workshop on Clustering Large data sets at the Third IEEE International Conference on Data Mining, Melbourne, Florida, USA.

Day, N.E. (1969). Estimating the Components of a mixture of normal distributions. *Biometrika, 56*, 463–474.

Dean, N., Murphy, T.B. & Downey, G. (2004). *Using unlabelled data to update classification rules with applications in food authenticity studies* (No. Technical Report 444). Department of Statistics, University of Washington.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood for incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39,* 1-38.

Domingos, P. & Hulten, G. (2001). A general method for scaling up machine learning algorithms and its application to clustering. In C.E. Brodley & A.P. Danyluk (Eds.), *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 106–113). San Francisco, CA: Morgan Kaufmann.

Domingos, P. & Hulten, G. (2002). Learning from finite data in finite time. In T.G. Dietterich, S. Becker & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14* (pp. 673-680). Cambridge, MA: MIT Press.

Domingos, P. & Hulten, G. (2003). A general framework for mining massive data streams. *Journal of Computational and Graphical Statistics,* 12(4), 945-949.

Evans, F.H., Alder, M.D. & DeSilva, C.J.S. (2003). *Determining the number of clusters in a mixture by iterative model space refinement - with application to free-swimming fish detection.* In C. Sun, H. Talbot, S. Ourselin & T. Adriaansen (Eds.), *Proceedings of the VIIth Biennial Australian Pattern Recognition Society Conference on Digital Image Computing: Techniques and Applications* (pp. 79-88). Collingwood, Vic.: CSIRO Publishing.

Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial data sets with noise. In E. Simoudis, J. Han & U. Fayyad (Eds.), *Proceeding of Second International Conference of Knowledge Discovery and Data Mining (KDD-96)* (pp. 226-231). Menlo Park, CA: AAAI Press.

Everitt, B.S. (1981). Contribution to the discussion of paper by M. Aitkin, D. Anderson & J. Hinder. *Journal of the Royal Statistical Society Series A, 144,* 457-458.

Everitt, B.S. (1988). A finite mixture model for the clustering of mixed-mode data. *Statistics and Probability Letters, 6*(5), 305-309.

Everitt, B.S. & Merette, C. (1990). The clustering of mixed-mode data: a comparison of possible approaches. *Journal of Applied Statistics, 17,* 283-297.

Fayyad, U. & Smyth, P. (1996). From massive data to science catalogs: applications and challenges. In J. Kettenring & D. Pregibon (Ed.), *Proceedings of the Workshop on Massive Data Sets* (pp. 129-142). Washington, DC: National Academy Press.

Figueiredo, M.A.T., & Jain, A.K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 381-296.

Fraley, C. & Raftery, A.E. (1998). How many clusters? Which clustering methods? Answers via model-based cluster analysis. *Computer Journal, 41,* 578-588.

Fraley, C. & Raftery, A.E. (1999). MCLUST: Software for model-based cluster analysis. *Journal of Classification, 16,* 297-306.

Fraley, C. (1999). Algorithms for model-based Gaussian hierarchical clustering. *SIAM Journal on Scientific Computing, 20*(1), 270-281.

Fraley, C. & Raftery, A.E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association, 97*(458), 611-631.

Fraley, C. & Raftery, A.E. (2003). Enhanced software for model-based clustering, density estimation, and discriminant analysis: MCLUST. *Journal of Classification, 20,* 263–286.

Fraley, C., Raftery, A.E. & Wehrens, R. (2005). Incremental Model-Based Clustering for Large Datasets with Small Clusters. *Journal of Computational and Graphical Statistics, 14,* 1-18.

Franco, J., Crossa, J., Villasenor, J., Taba, S. & Eberhart, S.A. (1998). Classifying genetic resources by categorical and continuous variables. *Crop Science, 38*(6), 1688–1696.

Franco, J., Crossa, J., Taba, S. & Eberhart, S.A. (2002). The modified location model for classifying genetic resources: II. Unrestricted variance-covariance matrices. *Crop Science, 42,* 1727-1736.

Franco, J. & Crossa, J. (2002). The modified location model for classifying genetic resources: I. Association between categorical and continuous variables. *Crop Science, 42,* 1719-1726.

Ganesalingam, S. & McLachlan G.J. (1978). The efficiency of a linear discriminant function based on unclassified initial samples. *Biometrika, 65*(3), 658-665.

Gersho, A. & Gray, R.M. (1992). *Vector Quantization and Signal Compression Communications and Information Theory*. Norwell, MA: Kluwer Academic Publishers.

Gionis, A., Mannila, H. & Tsaparas, P. (2007). Clustering aggregation. *ACM Transactions on Knowledge Discovery form Data (TKDD), 1*(1), Article 4.

Goodman, L.A. (1974a). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika, 61,* 215-231.

Goodman, L.A. (1974b). The analysis of systems of qualitative variables when some of the variables are unobservable. Part I: A modified latent structure approach. *American Journal of Sociology, 79*, 1179-1259.

Gordon, A.D. (1986). Links between clustering and assignment procedures. In F. De Antoni, N. Lauro & A. Rizzi (Eds.), *Proceedings in computational statistics: 7th symposium held at Rome 1986* (pp. 149-156). Heidelberg: Physica-Verlag.

Guha, S., Rastogi, R. & Shim K. (1998). CURE: An efficient clustering algorithm for large data sets. In A. Tiwary & M. Franklin (Eds.), *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (pp. 73-84). New York: ACM Press.

Halkidi, M. & Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representative. *Pattern Recognition Letters, 29*, 773-786.

Han, J. & Kamber, M. (2001). *Data mining: concepts and techniques*. New York: Morgan Kaufmann.

Hartigan, J. (1975). *Clustering Algorithms*. New York: Wiley-Interscience.

Heinen, T. (1996). *Latent Class and Discrete Latent Trait Models: Similarities and Differences*. Thousand Oaks, CA: Sage.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association, 58*, 13-30.

Hunt, L.A. & Jorgensen, M.A. (1999). Mixture model clustering using the Multimix program. *Australian and New Zealand Journal of Statistics, 41*, 153–171.

Ihaka, R. & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics, 5*(3), 299–314.

Jain, A.K., Duin, R.P.W. & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 22(1), 4–37.

Jamshidian, M. & Jennrich , R.I. (1993). Conjugate gradient acceleration of the EM algorithm. *Journal of the American Statistical Association, 88*, 221-228.

Jamshidian, M. & Jennrich , R.I. (1997). Acceleration of the EM algorithm by using Quasi-Newton methods. *Journal of the Royal Statistical Society, Series B, 59*, 569-587.

Jeffreys, H. (1935). Some tests of significance, treated by the theory of probability. *Mathematical Proceedings of the Cambridge Philosophy Society, 31(2)*, 203-222.

Jeffreys, H. (1961). *Theory of Probability*. Oxford: Oxford University Press.

Jin, H., Leung, K.S., Wong, M.L. & Xu, Z.B. (2005). Scalable model-based cluster analysis using clustering features. *Pattern Recognition, 38*, 637-649.

John, G.H. & Langley, P. (1996). Static versus dynamic sampling for data mining. In E. Simoudis, J. Han & U. Fayyad (Eds.), *Proceeding of Second International Conference of Knowledge Discovery and Data Mining (KDD-96)* (pp. 367–370). Menlo Park, CA: AAAI Press.

Jorgensen, M.A. & Hunt, L.A. (1996). Mixture model clustering of data sets with categorical and continuous variables. In D. L. Dowe, K. B. Korb & J. J. Oliver (Eds.), *Proceedings of the Conference on Information, Statistics and Induction in Science* (pp. 375-384). Singapore: World Scientific.

Karypis, G., Han, E.H. & Kumar, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer, 32*(8), 68-75.

Kass, R.E. & Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association, 90*(430), 773-795.

Kaufman, L. & Rousseeuw, P.J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley & Sons.

Keribin, C. (1998). Consistent estimate of the order of mixture models. *Comptes Rendues de l'Academie des Sciences, Série I - Mathématiques, 326*, 243–248.

Krzanowski, W.J. (1993). The location model for mixtures of categorical and continuous variables. *Journal of Classification, 10*, 25–49.

Lange, K. (1995). A Quasi-Newton acceleration of the EM algorithm. *Statistics Sinica, 5*, 1-18.

Lawrence, C.J. & Krzanowski, W.J. (1996). Mixture separation for mixed-mode data. *Statistics and Computing, 6*, 85-92.

Lazarsfeld, P.F. & Henry, N.W. (1968). *Latent Structure Analysis*. Boston: Houghton Mifflin.

Lee, S., Cheung, D. & Kao, B. (1998). Is sampling useful in data mining? A case in the maintenance of discovered association rules. *Data Mining and Knowledge Discovery, 2*, 232-262.

Leroux, M. (1992). Consistent estimation of a mixing distribution. *The Annals of Statistics, 20*, 1350–1360.

Liang, F., Mukherjee, S. & West, M. (2007). The use of unlabeled data in predictive modeling. *Statistical Science, 22*(2), 189-205.

Maitra, R. (2001). Clustering massive data sets with applications in software metrics and tomography. *Technometrics, 43*, 336-346.

McLachlan, G.J. (1982). The classification and mixture maximum likelihood approaches to cluster analysis. In P.R. Krishnaiah & L.N. Kanal (Eds.), *Handbook of Statistics* (Vol. 2, pp. 199-208). Amsterdam: North-Holland.

McLachlan, G.J. & Ganesalingam, S. (1982). Updating a discriminant function on the basis of unclassified data. *Communications in Statistics – Simulation and Computation, 11*(6), 753-767.

McLachlan, G.J. (1987). On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics, 36*, 318-324.

McLachlan, G.J. & Basford, K.E. (1988). *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker.

McLachlan, G.J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. New York: John Wiley & Sons.

McLachlan, G.J. & Krishnan, T. (1997). *The EM Algorithm and Extensions*. New York: John Wiley & Sons.

McLachlan, G.J. & Peel, D. (2000). *Finite Mixture Models*. New York: John Wiley & Sons.

McLachlan, G.J., Bean, R. & Ng, S.K. (2008). Clustering of microarray data via mixture models. In S. D. A. Biswas, J.P. Fine & M.R. Segal (Eds.), *Statistical Advances in Biomedical Sciences: Clinical Trials, Epidemiology, Survival Analysis, and Bioinformatics* (pp. 365-384). Hoboken, NJ: Wiley.

Meek, C., Thiesson, B. & Heckerman, D. (2002). The learning-curve sampling method applied to model-based clustering. *Journal of Machine Learning Research, 2*, 397-418.

Miller, R.G. (1986). *Beyond ANOVA: Basics of Applied Statistics*. New York: Wiley.

Mukerjee, S., Feigelson, E.D., Babu, G.J., Murtagh, F., Fraley, C. & Raftery, A.E. (1998). Three types of gamma ray bursts. *The Astrophysical Journal, 508*, 314-327.

Neal, R. & Hinton, G. (1998). A view of the EM Algorithm that justifies incremental, sparse, and other variants. In M. Jordan (Ed.), *Learning in Graphical Models* (pp. 355-371). Dordrecht: Kluwer Academic Publishers.

Ng, S.K. & McLachlan, G.J. (2003). On some variants of the EM algorithm for the fitting of finite mixture models. *Australia Journal of Statistics, 32*(1&2), 143-161.

Ng, S.K. & McLachlan, G.J. (2008). Expert networks with mixed continuous and categorical feature variables: a location modeling approach. In F. Columbus (Ed.), *Machine Learning Research Progress*. Hauppauge, New York: Nova.

Nigam, K., McCallum, A., Thrun, S. & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*, 103–134.

Olkin, I. & Tate, R.F. (1961). Multivariate correlation models with mixed discrete and continuous variables. *Annals of Mathematical Statistics, 32*, 448–465.

O'Neill, T.J. (1978). Normal discrimination with unclassified observations. *Journal of the American Statistical Association, 73*, 821–826.

Ordonez, C. & Omiecinski, E. (2002). FREM: Fast and robust EM clustering for large data sets. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (pp. 590-599). New York: ACM Press.

Palmer, C.R. & Faloutsos, C. (2000). Density biased sampling: An improved method for data mining and clustering. In W. Chen, J. F. Naughton & P.A. Bernstein (Eds.), *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 82-92). New York: ACM Press.

Peel, D. & McLachlan, G.J. (2000). Robust mixture modelling using the *t* distribution. *Statistics and Computing, 10*, 339-348.

Posse, C. (2001). Hierarchical model-based clustering for large datasets. *Journal of Computational and Graphical Statistics, 10*, 464-486.

Provost, F., Jensen, D. & Oates. T. (1999). Efficient progressive sampling. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 23–32). New York: ACM Press.

Rocke, D.M. & Dai, J. (2003). Sampling and subsampling for cluster analysis in data mining: With application to sky survey data. *Data Mining and Knowledge Discovery, 7*(2), 215-232.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6*, 461-464.

Shafer, J.C., Agrawal, R. & Mehta, M. (1996). SPRINT: A scalable parallel classifier for data mining. In T.M. Vijayaraman, A.P. Buchmann, C. Mohan & N.L. Sarda (Eds.), *Proceedings of the 22th International Conference on Very Large Data Bases (VLDB)* (pp. 544-555). Bombay: Morgan Kaufmann.

Shahshahani, B.M. & Landgrebe, D.A. (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing, 32,* 1087–1095.

Smith, A. & Spiegelhalter, D. (1980). Bayes factors and choice criteria for linear models. *Journal of the Royal Statistical Society, Series B, 42*, 213-220.

SPSS Inc. (2003). *Clementine 9.0 Algorithms Guide*.

Stanford, D.C. & Raftery, A.E. (2000). Principal curve clustering with noise. *IEEE Transactions on Pattern Analysis and Machine Analysis, 22*, 601-609.

Steiner, P.M. & Hudec, M. (2007). Classification of large data sets with mixture models via sufficient EM. *Computational Statistics and Data Analysis, 51,* 5416-5428.

Stephens, M. (2000). Dealing with label switching in mixture model. *Journal of the Royal Statistical Society, 62*, 795-809.

Taba, S., Diaz, J., Franco, J. & Crossa, J. (1998). Evaluation of Caribbean maize accessions to develop a core subset. *Crop Science, 38*, 1378-1386.

Taba, S., Diaz, J., Franco, J., Crossa, J. & Eberhart, S.A. (1999). *A core subset of LAMP, from the Latin American Maize Project, 1986-88* [CD-ROM]. Mexico DF: CIMMYT. Retrieved December, 2 2008 from

http://www.cimmyt.org/english/wps/publs/Catalogdb/catalog.cfm?data=20&monitor=3

Tantrum, J., Murua, A. & Stuetzle, W. (2002). Model-based clustering of large datasets through fractionization and refractionization. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 183-190). New York: ACM Press.

Thiesson, B., Meek, C. & Heckerman, D. (2001). Accelerating EM for large datasets. *Machine Learning, 45*(3), 279-299.

Xu, X., Ester, M., Kriegel, H.-P. & Sander, J. (1998). A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings of the Fourteenth International Conference on Data Engineering* (pp. 324-331). Washington, DC: IEEE Computer Society Press.

Wang, W., Yang, J. & Muntz, R. (1997). STING: A Statistical Information Grid Approach to Spatial Data Mining. In M. Jarke, M.J. Carey, K.R. Dittrich, F.H. Lochovsky, P. Loucopoulos & M.A. Jeusfeld (Eds.), *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB)* (pp. 186-195). San Francisco, CA: Morgan Kaufmann.

Ward, J. (1963). Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association, 58*, 236-244.

Wehrens, R., Simonetti, A.W. & Buydens, L.M.C. (2002). Mixture modelling of medical magnetic resonance data. *Journal of Chemometrics*, *16*, 274-282.

Wehrens, R., Buydens, L.M.C., Fraley, C. & Raftery, A.E. (2004). Model-based clustering for image segmentations and large datasets via sampling. *Journal of Classification*, *21*, 231-253.

Willse, A. & Boik, R.J. (1999). Identifiable finite mixtures of location models for clustering mixed-mode data. *Statistics and Computing*, *9*, 111-121.

Wolfe, J.H. (1967). *Normix: Computational methods for estimating the parameters of multivariate normal mixtures of distributions*. Unpublished manuscript.

Wolfe, J.H. (1970). Pattern clustering by multivariate mixture analysis. *Multivariate Behavioral Research, 5,* 329-350.

Wolfe, J.H. (1971). *A Monte Carlo study of the sampling distribution of the likelihood ratio for mixture of multinormal distributions* (No. NPTRL-STB-72-2). San Diego CA: Naval Personnel and Training Research Laboratory.

Zhang, T., Ramakrishnan, R. & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. In H.V. Jagadish & I.S. Mumick (Eds.), *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (pp. 103-114). New York: ACM Press.