Statistical Learning in Drug Discovery via Clustering and Mixtures

by

Xu Wang

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Statistics

Waterloo, Ontario, Canada, 2007

© Xu Wang, 2007

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

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

Abstract

In drug discovery, thousands of compounds are assayed to detect activity against a biological target. The goal of drug discovery is to identify compounds that are active against the target (e.g. inhibit a virus). Statistical learning in drug discovery seeks to build a model that uses descriptors characterizing molecular structure to predict biological activity. However, the characteristics of drug discovery data can make it difficult to model the relationship between molecular descriptors and biological activity. Among these characteristics are the rarity of active compounds, the large volume of compounds tested by high-throughput screening, and the complexity of molecular structure and its relationship to activity.

This thesis focuses on the design of statistical learning algorithms/models and their applications to drug discovery. The two main parts of the thesis are: an algorithm-based statistical method and a more formal model-based approach. Both approaches can facilitate and accelerate the process of developing new drugs. A unifying theme is the use of unsupervised methods as components of supervised learning algorithms/models.

In the first part of the thesis, we explore a sequential screening approach, Cluster Structure-Activity Relationship Analysis (CSARA). Sequential screening integrates High Throughput Screening with mathematical modeling to sequentially select the best compounds. CSARA is a cluster-based and algorithm driven method. To gain further insight into this method, we use three carefully designed experiments to compare predictive accuracy with Recursive Partitioning, a popular structure-activity relationship analysis method. The experiments show that CSARA outperforms Recursive Partitioning. Comparisons include problems with many descriptor sets and situations in which many descriptors are not important for activity.

In the second part of the thesis, we propose and develop constrained mixture discriminant analysis (CMDA), a model-based method. The main idea of CMDA is to model the distribution of the observations given the class label (e.g. active or inactive class) as a constrained mixture distribution, and then use Bayes' rule to predict the probability of being active for each observation in the testing set. Constraints are used to deal with the otherwise explosive growth of the number of parameters with increasing dimensionality. CMDA is designed to solve several challenges in modeling drug data sets, such as multiple mechanisms, the rare target problem (i.e. imbalanced classes), and the identification of relevant subspaces of descriptors (i.e. variable selection).

We focus on the CMDA1 model, in which univariate densities form the building blocks of the mixture components. Due to the unboundedness of the CMDA1 log likelihood function, it is easy for the EM algorithm to converge to degenerate solutions. A special Multi-Step EM algorithm is therefore developed and explored via several experimental comparisons. Using the multi-step EM algorithm, the CMDA1 model is compared to model-based clustering discriminant analysis (MclustDA). The CMDA1 model is either superior to or competitive with the MclustDA model, depending on which model generates the data. The CMDA1 model has better performance than the MclustDA model when the data are high-dimensional and unbalanced, an essential feature of the drug discovery problem!

An alternate approach to the problem of degeneracy is penalized estimation. By introducing a group of simple penalty functions, we consider penalized maximum likelihood estimation of the CMDA1 and CMDA2 models. This strategy improves the convergence of the conventional EM algorithm, and helps avoid degenerate solutions. Extending techniques from Chen et al. (2007), we prove that the PMLE's of the two-dimensional CMDA1 model can be asymptotically consistent.

Acknowledgements

First, I would like to express my deepest and sincere gratitude to my supervisor: Dr. Hugh A. Chipman for his guidance, insights, encouragement and patience. From him, I have learned not only how to become a researcher, but also how to mentor and interact with students. It is him who lead me step by step to finish my thesis, and helped me to decide my future career.

Second, I am grateful to my thesis committee members. Prior to moving to the University of British Columbia, Dr. William J Welch was also my supervisor in the first two years of my Ph.D. study. He showed me the interesting topic of drug discovery, and always gave me remarkable suggestions. Especially during the period of editing and correcting the CSARA paper (based on Chapter 2), he has provided me many meaningful comments, which improved the paper. I want to thank Dr. Jiahua Chen, who introduced a very interesting topic to me, Penalized Maximum Likelihood Estimation, and generously allowed me to read his manuscript on this topic. His suggestion motivated me to apply similar techniques to my model. He also spared his valuable time to answer my questions and read my first draft of the PMLE chapter. I also want to thank Dr. Mu Zhu for many interesting conversations we had and constructive comments on my research. I want to thank Dr. Shoja'eddin Chenouri for being my committee member and being a friend. I would like to thank my external examiner and thesis reviewer Dr. Rafal Kustra (Faculty of Medicine, University of Toronto) for his careful reading of the thesis, constructive comments and suggestions on my work. I also would like to thank Dr. Forbes Burkowski (School of Computer Science, University of Waterloo) for serving as a University examiner at my defense.

I greatly appreciate the help from Dr. S. Stanley Young on many drug discovery

topics. He has given me great encouragement and support for my drug research study.

Next, I want to give my special thanks to the faculty and staff in the department of Mathematics and Statistics, Acadia University. The department has provided a very good research environment. The university allowed me to access the library and computer network. The faculty and staff were very hospitable and made me feel at home in Wolfville.

The faculty in the department of Statistics and Actuarial Science, University of Waterloo, are great. They are knowledgeable and approachable. Special thanks to Dr. Richard Cook and Dr. Grace Yi Yun for sharing their personal research experience with me. I am also indebted to the staff members in the department. Special thanks to Mary Lou Dufton, who helped my registration and all the paper work when I was off-campus at Acadia University.

My special thanks also go to my teammates: Xianlin Ma, Sofia Mosesova, Hui Shen, Wanhua Su, Yuanyuan (Marcia) Wang, Yan Yuan, and Longyang Wu for our good time of discussion and collaboration. Special thanks to Sofia Mosesova, who always encouraged me to do better jobs.

Last, but definitely not least, I would like to thank my husband, Joel B. Price, for accompanying me through hard and difficult time during my study.

Finally, I want to dedicate my thesis to my dear family for their steadfast and unconditional love!

Xu Wang

To my family

Contents

List of Tables

List of Figures

Abbreviations

AHR	Average Hit Rate
CMDA	Constrained Mixture Discriminant Analysis
CMDA1	Constrained Mixture Discriminant Analysis First Order Model
CMDA2	Constrained Mixture Discriminant Analysis Second Order Model
CSARA	Cluster Structure-Activity Relationship Analysis
EM	Expectation-Maximization Algorithm
HTS	High-Throughput Screening
LDA	Linear Discriminant Analysis
MclustDA	Model-based Clustering Discriminant Analysis
MDA	Mixture Discriminant Analysis
MLE	Maximum Likelihood Estimate
MSE	Mean Squared Error
NCI	National Cancer Institute
PMLE	Penalized Maximum Likelihood Estimate
QDA	Quadratic Discriminant Analysis
QSAR	Quantitative Structure-Activity Relationship
RP	Recursive Partitioning
RDA	Regularized Discriminant Analysis
SAR	Structure-Activity Relationship

Glossary of Terms Used in Chapters 3-6

$i = 1, \ldots, n$	indexes observations in the sample
$y_i \in \{1, 2, \ldots, K\}$	a class indicator or response variable for observation i
$k = 1, \ldots, K$	indexes the K classes
$j = 1, \ldots, J_k$	indexes components in each class. J_k is the total number of components in k^{th} class
n_k	the number of observations in class k
$\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T)^T$	the observed data matrix
$\mathbf{z} = (\mathbf{z}_1^T, \ldots, \mathbf{z}_n^T)^T$	associated component-label
$\mathbf{x_c} = (\mathbf{x}^T, \mathbf{z}^T)^T$	the complete data
τ_k	the prior probability of class k
π_{jk}	the mixing proportion of the j^{th} component of class k
$\boldsymbol{\Psi_k}$	all the class-specific parameters in class k
$\boldsymbol{\Psi_G}$	all the global parameters in the model
$\boldsymbol{\Psi}$	all the parameters, i.e. $\boldsymbol{\Psi} = (\boldsymbol{\Psi_1}, \ldots, \boldsymbol{\Psi_K}, \boldsymbol{\Psi_G})$
$\boldsymbol{\Psi_0}$	the true parameters
$\hat{\boldsymbol{\Psi}}^{(m)}$	the value of $\boldsymbol{\Psi}$ after the m^{th} EM iteration
$\bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}}$	all the class-specific parameters in the j^{th} component of class k
$\bar{\boldsymbol{\Phi}}_{\boldsymbol{l}}$	the global parameters of the l^{th} dimension
$\boldsymbol{\Phi_{jk}}$	all the parameters in the density of the j^{th} component of class k, i.e. $\boldsymbol{\Phi_{jk}} = \{\bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}}, \bar{\boldsymbol{\Phi}}_{\mathbf{1}}, \ldots, \bar{\boldsymbol{\Phi}}_{\boldsymbol{P}}\}$
$\boldsymbol{S}$	sum of squares crossing all classes
$\boldsymbol{S_k}$	sum of squares in class k
$\boldsymbol{\Sigma_k} = \lambda_k \boldsymbol{D_k A_k D_k}^T$	the eigenvalue decomposition of the covariance matrix $\boldsymbol{\Sigma_k}$ of class k

λ_k	specifies the volume of density contours of class k
$\boldsymbol{A_k}$	diagonal matrix of eigenvalues, specifies the shape of class k
$\boldsymbol{D_k}$	eigenvectors, specifies the orientation of class k
$l_n(\mathbf{\Psi})$	incomplete-data log likelihood
$l_c(\mathbf{\Psi})$	complete-data log likelihood
D_{jk}	a tuning parameter in the penalty function for the class-specific σ_{jk} or $\mathbf{\Sigma_{jk}}$
D	a common tuning parameter
S_{jk}	the sample variance along x_j for the observations in class k between the first and the third sample quantiles
$\boldsymbol{S_{jk}}$	the mode of the prior distribution for the covariance matrix $\mathbf{\Sigma_{jk}}$ in the CMDA2 model
$f(\mathbf{x};\mathbf{\Psi})$	a parametric density function
$f(\mathbf{x};\mathbf{\Psi_k})$	a density function specific to class k
$f(\mathbf{x};\mathbf{\Phi_j})$	a density function of the j^{th} component
$f(\mathbf{x};\mathbf{\Phi_{jk}})$	a density function specific to the j^{th} component of class k
$\phi(\cdot)$	a standard normal density function with mean 0 and variance 1
$N(x;\mu,\sigma)$	a univariate normal density function with mean μ and variance σ
$MVN(\mathbf{x};\mu,\mathbf{\Sigma})$	a multivariate normal density function with mean vector μ and covariance matrix $\mathbf{\Sigma}$
$p_n(\mathbf{\Psi})$	penalty function on all the parameters, $\mathbf{\Psi}$
$\tilde{p}_n(\sigma)$	penalty function on one parameter, σ
$Pl_n(\mathbf{\Psi})$	penalized log likelihood, which equals $l_n(\mathbf{\Psi}) + p_n(\mathbf{\Psi})$
$l_{pc}(\mathbf{\Psi})$	penalized complete data log likelihood
$\tilde{\mathbf{\Psi}}_n = \arg\max_{\mathbf{\Psi}} Pl_n(\mathbf{\Psi})$	the penalized maximum likelihood estimator
$P_1(\mathbf{\Psi})$	the non-Bayesian penalty function for the CMDA1 model
$P_2(\mathbf{\Psi})$	the Bayesian penalty function for the CMDA1 model

Chapter 1

Drug Discovery and Data Sets

1.1 Drug Research

Drug discovery is a multidisciplinary endeavor occurring at the interface of biology, chemistry, computer science, statistics and informatics. Its history can be traced back over one hundred years. Drug research began when chemistry had reached a degree of maturity that allowed its principles and methods to be applied to problems outside of chemistry, and also when pharmacology had become a well-defined scientific discipline.

1.1.1 The Evolution of Drug Discovery

By the 1870's, drug research was affected heavily by some of the essential foundations of chemical theory, such as Avogadro's atomic hypothesis, the periodic table of elements, the theory of acids and bases, and especially August Kekule's pioneer-

ing theory on the structure of aromatic organic molecules. Analytical chemistry, in particular the isolation and purification of the active ingredients of medicinal plants, also demonstrated its value for medicine in the 19th century. Between 1871 and 1918, as a series of new institutions were created to support interdisciplinary drug research and development, a new way of finding, characterizing, and developing medicines led to the formation of a new industry, i.e. drug discovery (Drews 2000).

During the first half of 20th century, drug research was shaped and enriched by several new technologies, all of which left their imprint on drug discovery and on therapy (Drews 2000). For instance, microbiology, biochemistry and pharmacology helped shape the course of drug discovery and bring it to a level where new drugs are no longer generated solely by the imagination of chemists but result from a dialogue between biologists and chemists. Also the main effect of molecular biology for drug discovery lies in the potential to understand disease processes at the molecular or genetic level and to determine the optimal molecular targets for drug intervention.

1.1.2 High-throughput Screening

With the advent of genomic sciences, rapid DNA sequencing, combinatorial chemistry and cell-based assays, drug discovery has entered into a new period. In this new period, the critical problem is an ever-increasing number of targets and compounds. Pharmaceutical companies participating in drug discovery measure (assay) the activity of various chemical compounds against a biological target (e.g. a disease). With recent scientific and technological advances, such as larger chemical

libraries (chemical groupings of compounds) and robotic systems, assays of tens of thousands of compounds can be performed in a single day. This process is known as high-throughput screening (HTS).

In the HTS process, a number of compounds are screened against a given target and the compounds showing the biggest positive effect (e.g. hindering the development of diseases) are carried forward for more detailed analysis. Compounds with a positive effect above a predetermined threshold will be called active compounds or hits. With the prospect of many potential targets, the efficient design of biochemical assays is increasingly important. Methods for choosing compounds are an essential part of this design process. Also HTS creates new opportunities for structure-activity relationship (SAR) analysis and increases the need for effective statistical methods to identify trends and relationships in the data. Further, although the cost of testing a single chemical compound against a biological target is small, testing hundreds of thousands of compounds can become quite costly. Hence, sequential screening has been developed to help reduce costs and create a more efficient strategy to determine SAR models. Sequential screening will be described in the next section.

1.1.3 The Sequential Screening Paradigm

In today's drug discovery, HTS is unable to screen all possible compounds as the estimated number of possible drug molecules is roughly 10^{40} (Valler & Green 2000).

Hence sequential screening (Engels & Venkatarangan 2001) has been developed to help reduce costs and make HTS more efficient. Sequential screening combines

HTS and virtual screening (a screening model) in one integrated screening process. Instead of testing an entire chemical library against a biological target, only a fraction of the library, known as the initial sample set, is assayed. The purpose of the initial sample set is not to find as many actives as possible, but simply to collect data on a diverse set of compounds in the chemical space, so that a computational analysis of these data can identify trends and help to select a further set of compounds to be screened.

The sequential screening process is described in Figure 1.1. The flow of the process is as follows: The experiment starts with the initial sample of compounds, which is run through the HTS process. A quantitative structure-activity relationship (QSAR) analysis of the data is performed to identify descriptors (variables that quantify the structure of molecules) relevant to the biological activity. In this step, a model capable of predicting activity using descriptor values is fit using the data from the initial set screened. On the basis of the first QSAR, the whole data inventory is virtually screened in order to get a more focused set of compounds (i.e. the compounds predicted to be active) for a second round of HTS. Virtual screening of compounds consists of using the fitted model to predict activity for the untested compounds. Depending on the success of the HTS on the focused set, the project budget, and the available resources, one or more iterations of the HTS $\rightarrow$ QSAR $\rightarrow$ virtual screen cycle may be undertaken before the final QSAR analysis.

The QSAR analysis in the sequential screening process is a supervised learning problem, which uses both descriptors and biological activity of compounds to learn a predictive model. In Section 1.2, several descriptor sets will be introduced.

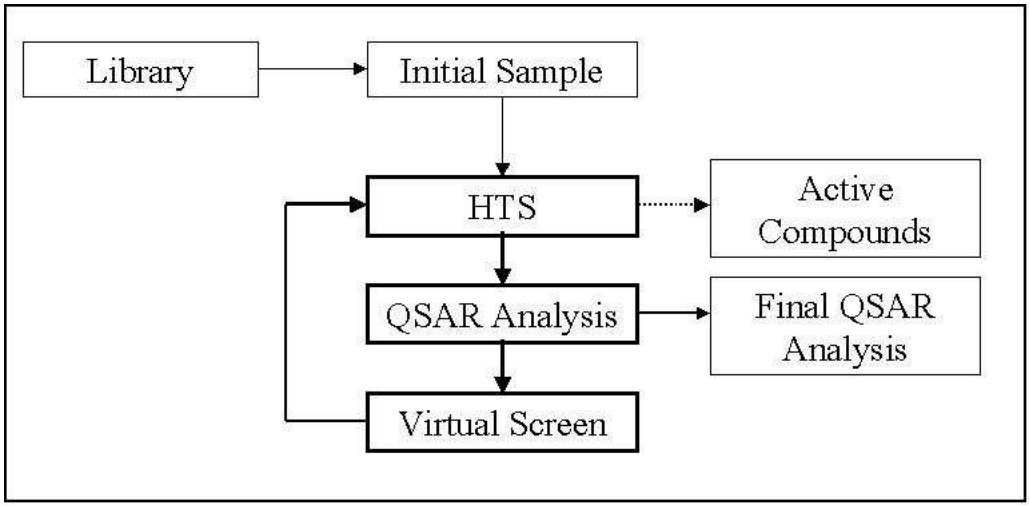

Figure 1.1: The Sequential Screening Paradigm.

1.2 Chemical Descriptors and Data Sets Used in the Thesis

Computational chemists have been able to qualify a compound's structure with many different sets of descriptors. Todeschini & Consonni (2000) list many of the available descriptors in their *Handbook of Molecular Descriptors*. Calculating descriptors is far less expensive than assaying the entire library of compounds, and as long as a set of descriptors can be generated, even compounds that do not yet exist or are not part of the company's chemical library can be virtually screened. In this section, we focus on introducing two drug discovery assays and five descriptor sets used in the thesis.

Assay			Descriptor Variables		
Name	Measurement	# Compounds	Set	#	Source
Yeast	binary	75,873	BCUT	6	GlaxoSmithKline
	continuous	75,873	BCUT	6	GlaxoSmithKline
AIDS	binary	29,812	BCUT	6	GlaxoSmithKline
	binary	29,374	BCUT	64	Feng et al. (2003)
	binary	29,374	Constitutional (CON)	46	Feng et al. (2003)
	binary	29,374	Property (PROP)	212	Feng et al. (2003)
	binary	29,374	Topological (TOP)	261	Feng et al. (2003)

Table 1.1: Data sets used in the thesis.

1.2.1 Data Sets

Two assays are used in the thesis with a variety of descriptor sets, as summarized in Table 1.1.

The first assay is from the National Cancer Institute (NCI) Yeast Anticancer Drug Screen (Simon, Dunstan, Lamb, Evans, Cronk & Irvine 2000), and will be referred to here as the Yeast data. The Yeast assay measures inhibition of human tumor cancer growth. For a tumor to develop, there must be a series of mutations, which cause cells to multiply uncontrollably. Because of the high degree of functional homology in biological activity between yeast and mammalian cells, many mutations can be modeled in yeast. Simon et al. (2000) carried out a screen of over 100,000 compounds from the repository at the NCI's Developmental Therapeutics Program to identify compounds that can inhibit the growth of the mutated cells. Hence, the Yeast assay data measure the percentage growth inhibition of the assayed compounds. Among 100,000 compounds screened, 75,873 are used in the

final analysis after three stages of screening. For most analyses we will convert the percentage inhibition to a binary inactive/active response. Of the 75,873 compounds, 6,834 are considered active, because they have growth inhibition of at least 70% (Simon et al. 2000); the proportion of active compounds is 8.2%. The Yeast data can be downloaded from `http://dtp.nci.nih.gov/yacds` (accessed April 26, 2003).

The second assay, from the NCI Developmental Therapeutics Program, relates to the HIV/AIDS virus and will be called the AIDS data. A description of this assay is provided by Lam, Welch & Young (2002). The original AIDS data (about $32,000$ compounds) can be downloaded from `http://dtp.nci.nih.gov/docs/aids/aids_data.html` (accessed April 26, 2003). Some observations with poor structure representations that are usually considered as non-drug candidates have been deleted from the original data (Lam et al. 2002), leaving us about $29,812$ compounds in the data. Feng, Lurati, Ouyang, Robinson, Wang, Yuan & Young (2003) used a slightly different assay data set, which has $29,374$ compounds. In the thesis, we use both of these assays. The biological activity is the amount of protection a compound gives to human CEM (immunofluorescence and cryoimmunoelectron microscopy) cells from HIV-1 infection. Two assay classifications, "moderately active" and "confirmed active", have been combined to form an "active" class (Lam 2001). There are approximately 2% actives.

The data for both assays were generated by HTS. When converted to a binary outcome, the Yeast data have a higher proportion of active compounds than found in the AIDS data. In general, these compounds are representatives of those in

pharmaceutical data sets. Although they are not completely typical as they include "toxic" compounds, which may selectively kill cancer cells, they should still provide useful information on the effectiveness of the QSAR sampling/analysis strategies.

In all experiments, we will treat the available data as if they were a compound library, and pretend to observe activity for selected subsets of the library. This will enable us to simulate the sequential screening process summarized in Figure 1.1.

Various descriptor sets, as summarized in Table 1.1, will be used to characterize chemical structure in the QSAR modeling. The next section describes these descriptor sets.

1.2.2 Chemical Descriptor Sets

Throughout the thesis, we shall assume that all descriptors are numeric variables, rather than categorical. Below we list various descriptor sets used in subsequent chapters.

6 BCUT descriptors

There are mainly three persons or research groups that have contributed to the evolution of BCUT descriptors. Burden (1989) originally suggested constructing a modified connectivity matrix to represent the hydrogen-suppressed connection table of the molecule. To build this connectivity matrix, the atomic numbers of the elements were put on the diagonal and values describing bond-type of each pair of atoms are put on the off-diagonal. The aim of his work was to produce descriptors that are highly compact but with minimal redundancy.

Based on the assumption that the smallest eigenvalues contain contributions from all the atoms and thus reflect topology of the molecule, the two smallest eigenvalues of this matrix were used as chemical descriptors of the molecule. The essence of the method is to solve the eigenvalue equation

$$\boldsymbol{BV} = \boldsymbol{Ve}, \tag{1.1}$$

where $\boldsymbol{B}$ is a real symmetric connectivity matrix to be defined. $\boldsymbol{V}$ is a matrix of eigenvectors, and $\boldsymbol{e}$ is a diagonal matrix of eigenvalues. Elements of the connectivity matrix are constructed so that the off-diagonal elements represent the strength of connection between atoms, and the diagonal elements the size of the elements. The rules defining $\boldsymbol{B}$ used in Burden's method are as follows:

(a) Hydrogen atoms are not included.

(b) The rows and columns of the connectivity matrix are arbitrarily numbered according to the heavy atoms.

(c) The diagonal elements of $\boldsymbol{B}$, B_{ii}, are the atomic numbers of the atoms.

(d) The element of $\boldsymbol{B}$ connecting atoms i and j, B_{ij}, is 0.1 for a single bond, 0.2 for a double bond, 0.3 for a triple bond, and 0.15 for an aromatic delocalized bond.

(e) Elements of $\boldsymbol{B}$ corresponding to bonds to terminal atoms (i.e., atoms with one connection only) are augmented by 0.01.

(f) All other elements of $\boldsymbol{B}$ are set at 0.001.

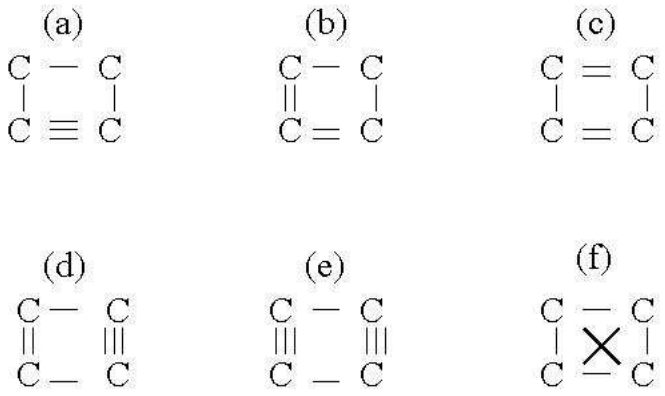

Figure 1.2: Some fragment structures for formula C_4H_4.

According to the above rules, the different fragment structures with the same formula can have different eigenvalues. For example, Figure 1.2 illustrates several fragment structures for a formula C_4H_4. The first two lowest eigenvalues for these fragment structures are given in Table 1.2. It is clear that even though these fragments have the same formula C_4H_4, but they have different eigenvalues due to their different molecular structures. Figure 1.3 shows the scatter plot of these eigenvalue descriptors in Table 1.2 for C_4H_4. It is interesting to see that (a) is close to (b); (d) is close to (e) and (f) is distant from all other compounds. From the structures of these fragment, we notice that (a) and (b), (d) and (e) are similar, while (f) is much more different from others. Therefore, the similar structures have close BCUT descriptors.

Burden's seemingly far-fetched ideas were successfully confirmed by Rusinko and Kipkus (A. Rusinko and A. H. Kipkus, unpublished result obtained at Chemical Abstract Service, Columbus OH) in 1993. They found structure searches based on Burden's suggestion were surprisingly comparable to the

	Smallest eigenvalue	Second smallest eigenvalue
(a)	5.6593	5.9407
(b)	5.6847	5.9990
(c)	5.7010	5.8990
(d)	5.6391	5.8609
(e)	5.6010	5.7990
(f)	5.9000	5.9000

Table 1.2: Descriptors for some C_4H_4 fragments.

results of accepted similarity searching procedures. Pearlman & Smith (1999) were inspired by the success of Rusinko and Kipkus, and extended Burden's approach of using one connectivity matrix to using multiple connectivity matrices. They proposed constructing three classes of matrices: one class with atomic charge-related values on the diagonal, a second class with atomic polarizability-related values on the diagonal, and a third class with H-bond-abilities on the diagonal. Also they put a variety of additional information on the off-diagonal including functions of inter-atomic distance, overlaps, computed bond-orders, etc. In addition to the smallest eigenvalue of each of the three connectivity matrices (as Burden suggested) Pearlman & Smith (1999) also used the largest eigenvalue of each matrix, which leave us in total 6 BCUT descriptors. The advantage of six BCUT numbers over other descriptors is their low dimensionality, which allows many statistical tools to be applied. However, they are not easy to interpret in terms of chemical structure.

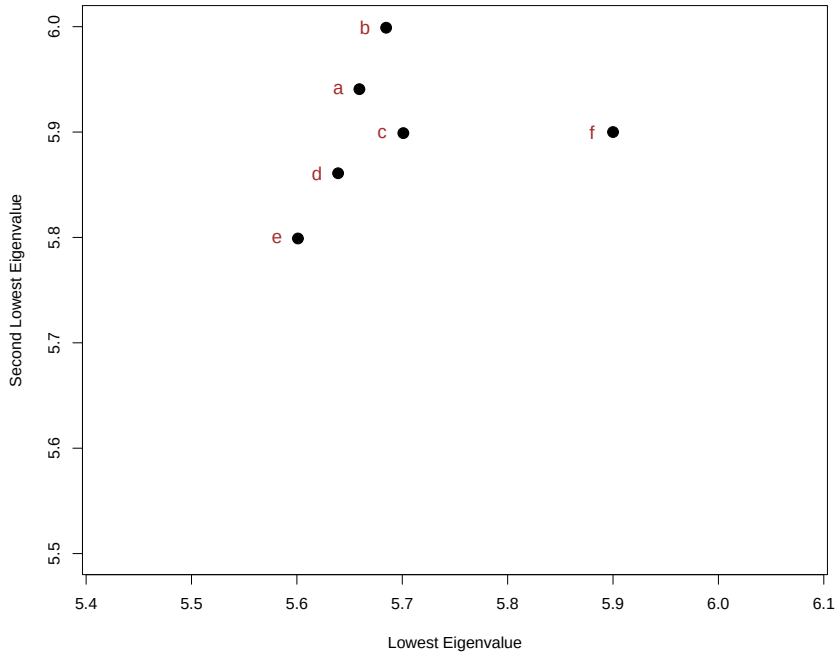

Figure 1.3: Scatter plot of the formula C_4H_4 using the descriptors in Table 1.2.

64 BCUT descriptors

Feng et al. (2003) considered four connectivity matrices presenting atomic properties — atomic mass, van der Waals volume, atomic electronegativity, and atomic polarizability. Instead of using the largest and smallest eigenvalues of each connectivity matrix, they used the eight largest and the eight smallest eigenvalues of each atomic property matrix, which give in total $16 \times 4 = 64$ BCUT descriptors.

46 **Constitutional descriptors (CON)**

The Constitutional descriptors are the measurements of the "constitution" of a compound. The 46 descriptors include information such as molecular

weight, atomic weight, atomic counts, etc. They depend only on the atoms in a molecule but not specifically on the connections between the atoms. For instance, the molecular weight of a compound is calculated as the sum of the atomic weight of all the atoms that make up one molecule of the compound without looking at their connectivity. Generally speaking, larger molecules will have higher values for all of the Constitutional descriptors. Many different molecules might have the same values of constitutional descriptors as there are many ways that the same atoms can be connected to make a valid molecule.

212 Property/Fragment Descriptors (PROP)

Property descriptors reflect physicochemical properties of molecules, like $\log P$ (the octanol-water partition coefficient, a measure of the hydrophobicity and hydrophilicity of a substance. In the context of drug-like substances, hydrophobicity is related to absorption, bioavailability, hydrophobic drug-receptor interactions, metabolism and toxicity), aromatic index, etc. They also include fragment descriptors, which indicate the kinds of fragments in a molecule and their frequencies. The fragments include atom/bond sequences and augmented atoms.

261 Topological Index (TOP)

The molecules are treated as topological objects where atoms become the vertices, and the bonds the edges of a molecular graph. The TOP descriptors, such as atomic order, relative eletronegativity, length of covalent radius, atomic mass, atomic and adjacent hydrogen mass, atomic polarity, atomic radius, and atomic eletronegativity etc, can be easily calculated. The TOP

descriptors can be used to evaluate structural similarity and diversity, making them widely used in QSAR analysis.

Ideally the descriptors should contain relevant information on the compounds and be few in number so that the subsequent analysis will not be too complex, but larger descriptor sets are very popular in drug discovery.

1.3 Challenges of Drug Discovery Data

The characteristics of drug discovery data generate many challenges for QSAR modeling:

1. Unbalanced response: although the data generated by HTS may have an enormous number of tested compounds, active compounds are often very rare.

2. Multiple mechanisms: the compound structure is complicated and may imply many mechanisms leading to activity.

3. Subspace-governed activity: the multiple mechanisms are usually determined by the low-dimensional subspaces of descriptors.

4. Nonlinear relationship: drug discovery data often involve threshold and non-linear effects when the chemical structure is represented by a set of descriptors.

5. Measurement errors: large random or systematic measurement errors may be present in the assays.

As an open research area, QSAR modeling attracts researchers on many statistical methods and techniques including simple methods such as linear regression, and more advanced methods like neural networks (NN), Partial Least Squares and recursive partitioning (RP, e.g. trees). Feng et al. (2003) built NN, PLS and RP using four different sets of chemical descriptors (64 BCUT variables, 46 Constitutional variables, 212 Property variables and 261 Topological variables in Table 1.1) and compared their performance.

It is well accepted that more complicated methods (e.g. NN, PLS and RP) should have more prediction power than linear regression to model a QSAR. Young & Hawkins (1998) applied a recursive partitioning procedure, FIRM, to a large, structure-activity data set and showed that different mechanisms of the data can be discovered. A later study (Wang 2005) using the same data set compared K-nearest neighbour classification (KNN), trees, neural networks, MARS (Friedman 1991), generalized additive models and logistic regression, concluding that KNN is one of the best methods.

Due to the difficulty of data and variety of the problems, QSAR modeling continues to inspire people to find better predictive models.

1.4 Structure of the Thesis

In this thesis, two QSAR models are developed. Chapter 2 describes an algorithm-based drug mining method: Cluster Structure-Activity Relationship analysis (CSARA). Comparisons between CSARA and Recursive Partitioning are conducted to evaluate the performance of CSARA. The second QSAR model is based on the idea of

mixture discriminant analysis, which is described in Chapter 3. The motivations for designing our special Constrained Mixture Discriminant Analysis (CMDA) model are also given in Chapter 3. In Chapter 4, the first order CMDA model (CMDA1) is discussed in detail. The degeneracy issue during the parameter estimation of the CMDA1 model is our focus. A Multi-step Expectation-Maximization (EM) algorithm is designed to handle the degeneracy problem. The other model-based solution to degeneracy, Penalized Maximum Likelihood Estimation (PMLE), is introduced in Chapter 5. The asymptotic consistency of the CMDA1 model is proved and confirmed by some simulation examples. Chapter 6 describes future research.

Chapter 2

CSARA: An Algorithm-based Drug Mining Method

2.1 QSAR Approaches: CSARA and RP

In theory, chemical compounds with similar structures will react with a biological target in a similar way (Lajiness 1997). Further, if the compounds with similar values of critical chemical descriptors can be grouped into one cluster, information about the activity of all compounds in the cluster may be obtained by simply assaying one or a few compounds randomly picked from the cluster. Engels & Venkatarangan (2001) suggested such a cluster-based approach for sequential screening experiments, namely Cluster Structure-Activity Relationship Analysis (CSARA). Even though Engels & Venkatarangan (2001) used several examples to show how useful the CSARA method is in the process of active compound selection compared to random selection, they did not compare CSARA with other QSAR

methods, like recursive partioning (RP). As RP is a very popular QSAR approach in drug discovery, it provides a good benchmark for comparison. In this chapter, we follow the same basic CSARA algorithm, but our focus is to deepen understanding of CSARA and explore its efficiency for screening drug data relative to RP.

The sequential screening framework described in Section 1.1.3 is applied here, with both CSARA and RP taking the role of supervised learning approaches. With the exception of Section 2.5, the response is assumed to be binary (active/inactive).

2.1.1 CSARA

The CSARA procedure is illustrated in Figure 2.1 and described as an algorithm below.

CSARA Algorithm

A. The entire compound library is available for sequential screening.

B. Compounds are clustered using descriptors and one compound is randomly selected from each cluster (shown in black in Figure 2.1(B)).

C. The randomly selected compounds, one from each cluster, are tested for biological activity. In Figure 2.1(C), the compounds shown in black are active and those in gray are inactive.

D. All the compounds in the clusters with an active compound from (C) are assayed to get more accurate estimation of their biological activities.

The CSARA algorithm is based on the belief that compounds with similar chemical structures react with targets in a similar way. In Figure 2.1, CSARA partitions the entire compound library into six clusters, and one representative of each cluster is randomly selected and tested. If the selected compound is active, we place all other compounds belonging to that cluster in the focused library for the second round of HTS. However, if the selected compound is inactive, this would suggest that the remaining compounds in the cluster are also inactive, and they are not included in the second HTS. Although the illustration in Figure 2.1 has only six clusters, CSARA would typically use hundreds or thousands of clusters.

In fact, CSARA is a two-stage sequential screening process. Steps B, C, and D in the CSARA algorithm correspond to the key boxes in Figure 1.1: CSARA's

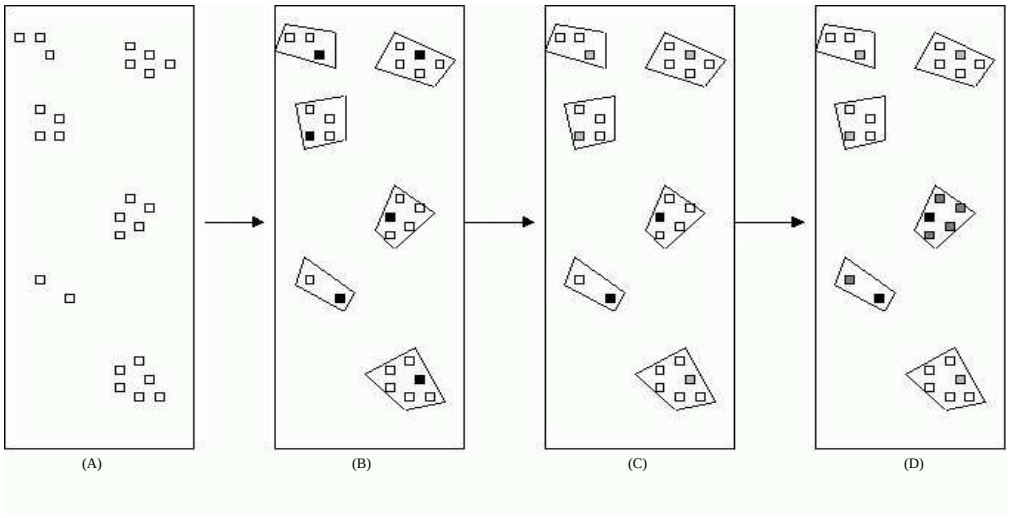

Figure 2.1: The CSARA process. Adapted from Engels and Venkatarangan (2001).

Step B is selecting an initial sample of compounds or training data; Step C is HTS to measure the activities of the initial sample; Step D combines QSAR analysis and virtual screening in order to get the focused library.

In Step B, the whole compound library is partitioned into groups by a clustering algorithm, using descriptors. The partitioned groups are then partly tested (here, only one compound each group) and used to determine the activities of untested compounds. CSARA uses both descriptors and biological activities of compounds to analyze the relationship between molecular structures and biological activity, so CSARA is a supervised learning algorithm.

K-means Algorithm

A critical part of CSARA is cluster analysis (Step B). Partitioning the entire compound library into clusters can be approached with a wide variety of clustering

algorithms (Dunbar 1997). Available algorithm-based methods of seeking clusters can be categorized broadly as hierarchical methods and partitioning methods. In this chapter, we use one popular partitioning method, K-means (MacQueen 1967).

For the K-means algorithm, K represents the required number of clusters that must be supplied by the user. In general, the K-means algorithm works as follows:

1. Randomly choose K unique compounds with descriptor vectors, $\mathbf{m_1}, \ldots, \mathbf{m_K}$, which serve as "centres" or "means" for the K clusters. Let $C^{(1)}, \ldots, C^{(K)}$ represent a partition of data indices $1, \ldots, n$ into clusters $1, \ldots, K$.

2. Iterate the following steps until the cluster centres do not change with an update:

   (a) Update cluster memberships $C^{(1)}, \ldots, C^{(K)}$ by allocating each compound to the cluster with the closest centre.

   (b) Recompute the K cluster centers, $\mathbf{m_1}, \ldots, \mathbf{m_K}$, by averaging the points within clusters:

   $$\mathbf{m_k} = \frac{1}{n_k} \sum_{i \in C^{(k)}} \mathbf{x_i} \quad for \quad k = 1, \ldots, K, \tag{2.1}$$

   where n_k is the number of points in $C^{(k)}$ and $\mathbf{x_i}$ is a vector representing descriptor variables of the ith point.

The aim of the K-means algorithm is to divide n points in p-dimensions into K clusters so that the within-cluster sum of squares is minimized. The within-cluster sum of squares criterion is

$$\min_{C^{(1)},\ldots,C^{(K)};\mathbf{m_1},\ldots,\mathbf{m_K}} \sum_{k=1}^{K} \sum_{i \in C^{(k)}} \|\mathbf{x_i} - \mathbf{m_k}\|^2. \tag{2.2}$$

where $\| \bullet \|^2$ denotes squared Euclidean distance. The K-means algorithm is not guaranteed to find the global minimum sum of squares (2.2). Instead, the K-means algorithm will find a local optimum. Hartigan & Wong (1979) propose an improved version of K-means such that the movement of a point from one cluster to another will not reduce the within-cluster sum of squares. Although Hartigan & Wong's algorithm does not guarantee a global optimum, it tends to find better local optima.

Compared to other clustering algorithms, K-means is a fast algorithm because distance calculations are only made from each point to the centers $\mathbf{m_1}, \ldots, \mathbf{m_K}$ rather than considering all pairwise distances. Also finding the nearest cluster for each compound via Euclidean distance is a fast calculation. Due to its very quick computability, K-means is an appealing method when dealing with large data sets, especially those arising from HTS.

There are several issues needed to be considered when K-means is applied.

(1) **Scaling**

The K-means algorithm usually uses Euclidean distance to determine the cluster center to which a compound is closest. The scaling of the descriptors does have an effect on Euclidean distances. For the purposes of this thesis, the descriptors were standardized by dividing them by their standard deviations.

(2) **Unique starting values**

A characteristic typical of HTS data can result in difficulty with the K-means algorithm. According to Young, Lam & Welch (2002), HTS is an imprecise exercise, which results in some compounds being assayed repeatedly. The replicated compounds in the data set are a potential difficulty in the application of the K-means

algorithm. The following scenario illustrates what can happen when replication is present. Suppose compound i is assayed twice. In step 1 of the K-means algorithm, it is possible to choose compound i to represent two different cluster centers. For simplicity, we say cluster 1 and cluster 2 have center i. In step 2 of the algorithm, the compound in cluster 1 will be reallocated to cluster 2 resulting in cluster 1 being empty. In this case, there are now only $K-1$ clusters instead of the desired K. To overcome this problem, only unique compounds should be selected as initial centers. Then the replicated compounds can be allocated to the same clusters in which their replicates reside.

(3) **Multiple runs**

Since the K-means algorithm gives different results for different starting values, multiple runs are needed to find better optima. The random selection of initial values make it likely that different runs will find different local optima.

2.1.2 Recursive Partitioning

Recursive Partitioning (RP) uses a tree-structured set of questions about the descriptor variables to recursively divide the data into groups in which the response variable is as homogeneous as possible. To build an RP model, the descriptor space is recursively subdivided into nodes of a tree. To identify the best split for a specific node, the algorithm considers all possible binary splits for each descriptor variable and chooses the optimal one by some criterion (Breiman, Friedman, Olshen & Stone 1984). This splitting is carried out recursively until some stopping condition is reached. Stopping criteria can be employed directly to choose tree size

(Hawkins & Kass 1982), or a large tree can be grown and then pruned (Breiman et al. 1984).

An Example

Recursive partitioning can be visualized as a tree. Each segment of the tree is called a node. Usually, a parent node is split into 2 descendant nodes. In Figure 2.2, checkboard training data (Welch 2002) and a tree grown to this data are plotted. The original node in which all of the data lie is called the root node. Class 0 has 40 of the 55 cases, i.e. 73%. If the tree growing algorithm is stopped at the root node, the root node would be classified as class 0, the majority class. The root node is split into two descendant nodes, which have $X_2 < 7.015$ and $X_2 \geq 7.015$, respectively. Ideally, the split should make each node as pure as possible. Here, the descendants do not show much improvement in purity of the response, so another split is needed before a big jump in purity appears.

The two descendants of the root node are themselves split to create their descendants. The two descendants of the root's left node, for example, are split based on the value of X_1: they have $X_1 < 6.39$ and $X_1 \geq 6.39$, respectively, with 10% and 62% class 1 objects, respectively. These nodes are closer to the ideal of all class 0 or all class 1. Neither of these needs further splitting; they are called terminal nodes. Similarly, the root's right descendant is split into two terminal nodes. Each terminal node is classified according to its majority class. Note that the two descendants of the root node are split using different cut-offs for X_1 depending on the value of X_2, indicating that the tree can include interaction effects automatically.

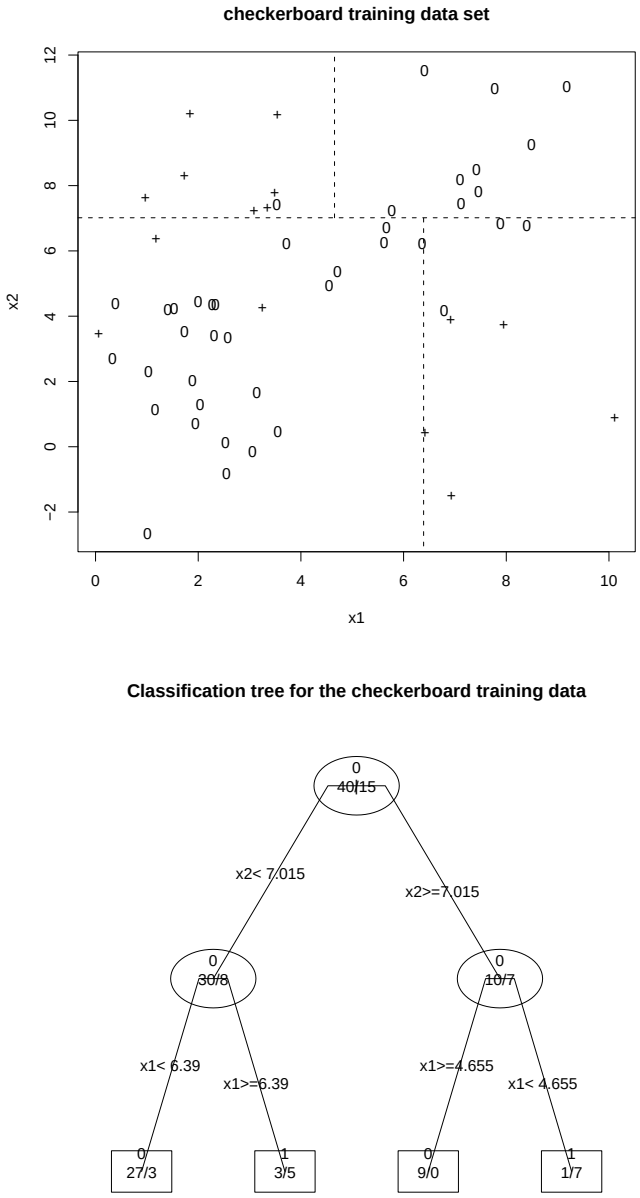

Figure 2.2: Top: The scatter plot and partitioning of the checkerboard explanatory variable space by the bottom tree. Class 0 and Class 1 training data are denoted by "0" and "+", respectively. Bottom: Classification tree for the checkerboard training data.

The top plot in Figure 2.2 also portrays the partitioning of the explanatory variable space of the checkerboard training data. Each rectangle corresponds to a terminal node. For example, the bottom-left rectangle corresponds to the first terminal node with 27 Class 0 and 3 Class 1 cases.

Tree Size Selection

Our research initially considered a pruning approach. When pruning a tree, a variety of criteria, including misclassification rate, the Gini index and entropy, can be employed. For unbalanced classification problems such as drug discovery, where one class (i.e. active class) is rare, misclassification rate is an inappropriate pruning criterion. This is because we desire a tree that can rank compounds by the probability that they will be active, rather than just classify them as active/inactive.

The performance measure we will use to assess tree performance is the hit rate, which is the percentage of hits among those compounds selected. In order to facilitate comparisons with CSARA, the number of compounds selected is matched in each experiment with the number selected by CSARA.

Ideally, a tree would be grown and pruned according to the hit rate. However, our goal is to compare "off-the-shelf" versions of tree growing algorithms with CSARA, and hit rate is not a standard criterion for growing or pruning a tree. Thus we choose the Gini index, $\sum_{i=1}^{n} \hat{p}_i(1-\hat{p}_i)$, where $\hat{p}_i$ is the predicted probability of activity for observation i, as a surrogate measure for tree growing and pruning. The Gini index is among the most appropriate measures since it encourages models to accurately predict p_i, the probability of activity for observation i, rather than just

predict a class label.

Once a sequence of pruned trees has been generated using the surrogate pruning criterion, predictions can be generated and the hit rate (our preferred performance measure) can be evaluated for all trees in the sequence. Thus a cross-validated measure of the hit rate will be obtained for each tree in the nested sequence of trees. We explore two different ways of generating this hit rate: either calculating a hit rate separately for each of the ten folds, and averaging them, or combining the predictions from all ten validation sets, then ranking the compounds and generating a single hit rate. Similar results were obtained using either strategy. All calculations were carried out in R (R Development Core Team 2006), using the `rpart` library (Therneau & Atkinson 2006).

In experiments with the AIDS Antiviral data with 64 BCUT descriptors (Section 1.2.1), we made two unexpected discoveries with respect to pruning and tree size:

1. Cross-validation, the most common method of selecting tree size, appears to select a tree with too few nodes in unbalanced-class problems.

2. The largest trees yield the best (or near-optimal) out-of-sample predictive accuracy.

In the remainder of this section, we outline the ideas behind these findings. A consequence is that for the remainder of the chapter, we choose a large tree size rather than use cross-validation, since this seems to produce the most competitive RP models.

To understand the two results, we first review cross-validation and cost-complexity pruning. Breiman et al. (1984) presented a cross-validated cost-complexity pruning approach to selecting the best tree size. Given a tree with size $\|T\|$ ($\|T\|$ represents the number of terminal nodes), the pruning criterion R_α is defined as:

$$R_\alpha = C(T) + \alpha\|T\| \tag{2.3}$$

where $C(T)$ is a smaller-the-better measure such as the Gini index and α is called the cost-complexity parameter. The term $\alpha\|T\|$ is a penalty for the size of the tree. Large values of α penalize big trees and lead to more pruning. Breiman et al. (1984) showed that as α increased, there exists a well defined nested sequence of pruned trees that optimize (2.3).

The tree with the best cross-validated cost-complexity would be chosen as follows: First, a large tree is grown using the training data. A nested sequence of m pruned trees (with $s_1 < s_2 < ... < s_m$ terminal nodes) is generated, minimizing a cost-complexity criterion. In fact, we choose the cost-complexity parameters to generate the nested sequence of trees. Since each cost-complexity parameter corresponds to a specific tree size s_i, the sequence of the tree sizes is used in the rest of the chapter for easier understanding. The goal is then to choose one tree from this nested sequence (that is, choose $s^* \in s_1, ...s_m$). This is accomplished by cross-validation. For example, with 10-fold cross-validation, 10 different large trees are grown and pruned, yielding 10 nested sequences, with sizes corresponding to $s_1, ..., s_m$. Each of the 10 different trees is grown using 90% of the training data, and holding out a different validation set of 10% of the training data. For each tree size s_i, prediction errors are averaged across the 10 validation folds, and the best

tree size s^* is chosen so as to minimize this cross-validation error.

For the AIDS assay data and 64 BCUT descriptors, we generated 20 training sets of 15,000 observations, about half of the data set, and for each set chose the optimal tree size according to cross-validated hit rate. In order to see whether we are choosing the right size, we "cheat" by also using the remaining (approximately 15,000) observations as a test set to choose the right tree size. The tree sizes chosen by these two strategies are displayed in Figure 2.3. Cross-validation typically selects a tree with between 1 and 100 terminal nodes, occasionally selecting a tree with around 150 terminal nodes. In contrast, the test set reveals that the best tree size is never below 60 terminal nodes, and is usually greater than 150 terminal nodes. The choice by cross-validation of a too-small tree results in a decrease in prediction accuracy for the test set, in comparison to the optimal tree size. Note that the use of a test set to choose tree size is generally inappropriate, and is used here simply to illustrate that cross-validation seems to select an inappropriate sized tree.

A possible reason for this discrepancy is the large number of "ties" that a tree produces in its predictions for the probability of activity. All observations falling in a specific terminal node of the tree will receive the same prediction. Tie structure among the predictions can affect predictive accuracy in unbalanced response problems where ranking is the goal. In cross-validation, since only 10% of the data are predicted by a tree corresponding to each fold, and the trees are slightly different for each fold, there are fewer ties in the predictions. This may make a small tree appear to be a better performer under cross-validation, where it seems to generate fewer ties, than for an independent test set, where one tree is

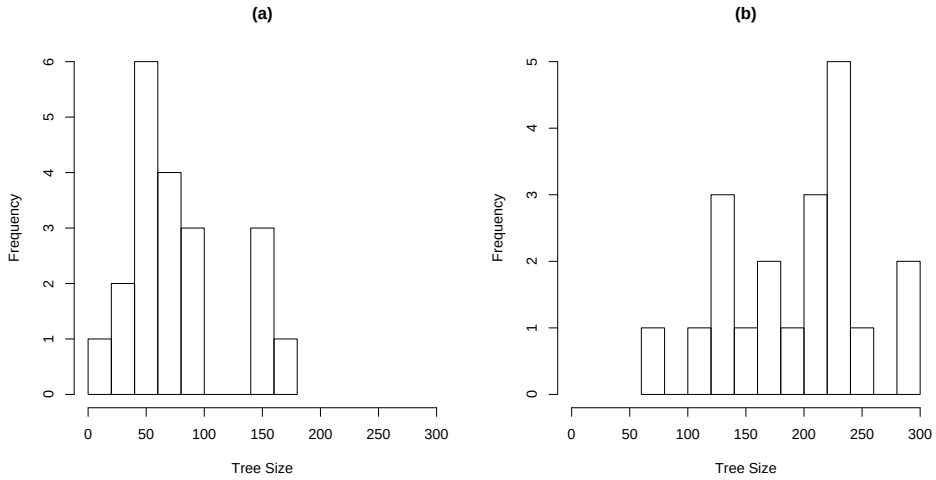

Figure 2.3: Histograms of the tree sizes chosen by (a) cross validation and (b) performance on an independent test set.

generated from all the training data, with more ties in predictions.

2.2 Evaluation Plan for CSARA and RP

In Section 2.2.1 we describe three strategies for data sampling/analysis based on CSARA, RP, and a hybrid method. We also discuss a performance metric for their evaluation.

2.2.1 Three Sampling/Analysis Strategies and Their Evaluation

All three strategies to be described will be evaluated in terms of their performance in identifying active compounds in the second round of HTS.

Specifically, we will use the hit rate (HR), which is the proportion of compounds identified as biologically active via an assay among a specified group of compounds selected for screening, i.e. active ÷ selected. As CSARA is a two-stage process of sequential screening, hits (active compounds) and hence the hit rate come from both the first and the second screens. A tree generated by RP, however, is used to make predictions and hence choose compounds for only the second screen. Thus, all comparisons of HR will be made only for the second screen. The assay results from the first screen are treated as "training" data, and those from the second screen are treated as "test" data. Nonetheless, the performance of RP will depend on the training data used to fit it. Thus, we will also consider the role of training data in defining strategies.

Two distinct methods are used to sample training data:

Methods for Sampling Training Data

Cluster: The compound collection is clustered into K clusters, and one compound is randomly chosen from each cluster. This is Step B of the CSARA algorithm, illustrated in Figure 2.1(B).

Random: K compounds are chosen completely at random (without replacement) from the collection.

The Cluster method is a component of CSARA, but it can also be used to generate training data for RP. It allows us to evaluate the differences between CSARA and RP when the same training data are used in modeling the QSAR. The Random method can be used only for RP. It will allow us to assess the usefulness

of a training sample designed to be diverse versus a random selection.

Hence, there are three different data sampling/analysis strategies:

Three Sampling/Analysis Strategies

CSARA: All steps of the CSARA method as described in Section 2.1.1.

Cluster/RP: The Cluster sampling method is used to select the initial sample set, an RP tree is trained on this set, and predictions from the tree model are used to choose the follow-up sample.

Random/RP: The Random sampling method is used to select the initial sample set, an RP tree is trained on this set, and predictions from the tree model are used to choose the follow-up sample.

The comparisons of interest are CSARA versus Cluster/RP and Random/RP, to understand the impact of the QSAR modeling method, and Cluster/RP versus Random/RP to understand the impact of the sampling of the training set.

2.3 Experiments and Results

2.3.1 Yeast Data

First, we apply CSARA, Cluster/RP and Random/RP to the Yeast data. We need to specify K, the number of clusters used to generate the first screen. Since one

	First Screen		Second Screen (CSARA)			Cluster/RP		Random/RP	
Trial	Hits	HR (%)	# compounds	Hits	HR (%)	Hits	HR (%)	Hits	HR (%)
1	243	8.1	5,648	874	15.5	819	14.5	676	12.0
2	250	8.3	5,720	861	15.1	870	15.2	753	13.2
3	261	8.7	6,060	841	13.9	678	11.2	788	13.0
4	262	8.7	6,272	1020	16.3	917	14.6	777	12.4
Ave.	254	8.47	5,925	899	15.2	821	13.9	749	12.6

Table 2.1: Results for the Yeast data and $K = 3,000$ clusters

compound from each cluster will be assayed, the resultant training set will have K observations to be used for training data: We take $K = 3,000$, 7,000 and 15,000. Because all methods considered rely on some form of randomization to select the training data, four trials are run at each of the three levels of K. The six BCUT descriptors described in Section 1.2 are used.

The results are presented in Tables 2.1–2.3. As the results follow similar patterns across values of K and the four trials, we explain in detail only the first row of Table 2.1. The first column is the trial index. The second and third columns correspond to the first screen using Steps A–C of the CSARA method. After the first screen, there are 243 hits in the training set, giving a hit rate of $243/3,000 = 8.1\%$. The fourth to sixth columns give results from the second HTS using Step D of CSARA: All compounds in the active clusters (not including those in the first screen) are treated as the test data for the second round HTS. In this example, 5648 compounds are in the test data, there are 874 hits, and consequently the hit rate is $874/5,648 = 15.5\%$.

The seventh and eighth columns give results from the second screen using Clus-

ter/RP. In the first row of Table 2.1, Cluster/RP selects the 5,648 compounds with highest predicted probabilities of being active (the size of the test set is matched to CSARA's). Roughly 819 or 14.5% are active. The same technique of selecting for the second screen is used for Random/RP in columns 9–10, but the training data are $K = 3,000$ randomly chosen compounds. The last row of Tables 2.1–2.3 gives average values across the four trials/test splits.

For tree models, calculation of the number of hits is complicated by the presence of many tied predictions. All test points falling in the same terminal node will receive the same predicted probability of activity. For example, suppose the best 14 nodes give us 5,448 compounds from the test set, and the 15^{th} best node has an additional 300 test set compounds. We need to select 200 of these 300 compounds to give the desired 5,648 compounds. We deal with this problem by reporting an expected number of hits under random sampling of 200 compounds from the 300 available in the node. This is equivalent to linear interpolation of number of hits between the two nodes.

In Tables 2.1–2.3, within each row, CSARA and Cluster/RP can be compared since they have the same training data set. Comparing Random/RP with CSARA or Cluster/RP is not meaningful within a row because they have different training sets. Comparisons using the "average" row are valid between all three methods.

The results given in Tables 2.1–2.3 can be summarized as follows:

- As K increases, the hit rate in the test set also increases. This makes intuitive sense because a larger training set gives more information to the tree and clustering algorithms, enabling them to uncover the QSAR within the HTS

	First Screen		Second Screen (CSARA)			Cluster/RP		Random/RP	
Trial	Hits	HR (%)	# compounds	Hits	HR (%)	Hits	HR (%)	Hits	HR (%)
1	584	8.3	5,783	1,056	18.3	888	15.4	960	16.6
2	584	8.3	5,785	1,069	18.5	938	16.2	857	14.8
3	580	8.3	5,515	1,023	18.6	872	15.8	809	14.7
4	583	8.3	5,735	1,053	18.4	890	18.4	877	15.3
Ave.	583	8.3	5705	1,050	18.4	897	16.4	876	15.3

Table 2.2: Results for the Yeast data and $K = 7,000$ clusters

	First Screen		Second Screen (CSARA)			Cluster/RP		Random/RP	
Trial	Hits	HR (%)	# compounds	Hits	HR (%)	Hits	HR (%)	Hits	HR (%)
1	1,182	7.9	4,771	1,066	22.3	898	18.8	845	17.7
2	1,321	8.8	5,417	1,110	20.5	928	17.1	955	17.6
3	1,259	8.4	5,197	1,064	20.5	871	16.8	894	17.2
4	1,286	8.6	5,099	1,057	20.7	871	17.1	860	16.9
Ave.	1,262	8.4	5,121	1074	21.0	892	17.5	889	17.4

Table 2.3: Results for the Yeast data and $K = 15,000$ clusters

	CSARA		Cluster/RP		Random/RP	
K	Mean HR	(se)	Mean HR	(se)	Mean HR	(se)
3,000	20.3%	(.6%)	15.6%	(.7%)	15.6%	(.8%)
5,000	24.5%	(.7%)	19.5%	(.8%)	18.0%	(.6%)
7,000	26.4%	(.4%)	21.3%	(.8%)	20.6%	(.5%)
10,000	31.6%	(.5%)	26.7%	(.7%)	22.5%	(.5%)
15,000	36.6%	(.5%)	30.0%	(.5%)	24.7%	(.5%)

Table 2.4: Results for the AIDS data

data. Also, as K increases the change in the hit rate is larger for CSARA than for Random/RP and Cluster/RP.

- Whatever K is chosen, CSARA always has a higher hit rate than Cluster/RP and Random/RP. This indicates that CSARA outperforms Cluster/RP and Random/RP.

2.3.2 AIDS Data

A similar analysis is carried out for the AIDS data, but with $K = 3,000$, 5000, 7,000, 10,000 and 15,000, and 20 trials (training sets) are used. The descriptor set is six BCUTs and there are $29,812$ compounds.

Instead of presenting results for each trial, as in Tables 2.1–2.3, we report in Table 2.4 averages and standard errors across the 20 test sets. Since each row represents an average over multiple training sets, all three methods can be compared within a row.

Table 2.4 shows that:

- The standard errors of the hit rate are very small, implying that 20 trials are sufficient to compare the mean hit rates of the three sampling/analysis strategies.

- For all values of K considered, the mean hit rate of CSARA is consistently larger than that of Cluster/RP or Random/RP. Thus, CSARA method is competitive and efficient relative to RP for identifying active compounds here.

- Cluster/RP outperforms Random/RP, particularly for larger values of K: A diverse training set is advantageous here.

Formal statistical tests, i.e., paired t-tests for CSARA versus Cluster/RP and unpaired t-tests for the other comparisons, confirm the above findings at a 5% significance level. The differences in mean HR are statistically significant for CSARA against either RP competitor for all values of K, and for Cluster/RP against Random/RP for $K = 10,000$ and $15,000$.

2.3.3 Adding Irrelevant Descriptors

In order to test the stability of CSARA and RP methods, we carry out a new experiment on the AIDS data adding several irrelevant or "junk" descriptor variables to the six BCUT descriptors when modeling the AIDS data. The values of the first new, irrelevant descriptor are generated by randomly permuting the values of the first BCUT, the second irrelevant variable is generated in the same way from the second BCUT, and so on. Different permutations are used for each variable. We take $K = 15,000$ in this experiment.

# Irrelevant	CSARA		Cluster/RP		Random/RP	
Descriptors	Mean HR	(se)	Mean HR	(se)	Mean HR	(se)
0	36.6	(0.531)	30.0	(0.522)	24.7	(0.453)
1	26.2	(1.93)	28.7	(1.50)	24.1	(0.85)
2	20.5	(0.53)	26.8	(0.74)	23.8	(1.68)
3	14.0	(1.84)	23.7	(0.42)	22.8	(1.82)
4	12.1	(0.59)	23.2	(0.69)	25.7	(1.51)
5	9.5	(0.68)	23.6	(2.41)	21.3	(1.55)
6	8.2	(0.97)	19.5	(2.05)	20.5	(0.59)

Table 2.5: Mean hit rates and standard errors (%) for the AIDS data and $K = 15,000$ when irrelevant descriptors are added to the six BCUTs

Table 2.5 illustrates the effect of adding 1–6 junk variables. The hit rates reported are means over four trials. With more irrelevant descriptors, the mean hit rate of CSARA decreases much more quickly than that of Cluster/RP or Random/RP. Among these three methods, Random/RP is the most stable. RP has built-in variable selection due to choosing a variable at each split according to an optimality criterion (here the gini index). CSARA has no such capability, as all descriptors are included in the distance metric for clustering. Similarly, the benefit due to clustering in selection of a training set for RP diminishes with more irrelevant variables.

2.4 Experiments with High-dimensional Descriptor Sets

In order to understand how CSARA performs with higher-dimensional descriptor sets, we make comparisons between CSARA, Cluster/RP, and Random/RP using the AIDS assay data and four further descriptor sets. The four sets, summarized earlier in Table 1.1, are BCUT (64 variables), Constitutional (46 variables), Property (212 variables) and Topological (261 variables).

As before, we run 20 trials for different values of K and calculate the mean hit rates. Mean hit rates are plotted against K in Figure 2.4

Figure 2.4 shows that CSARA outperforms Cluster/RP and Random/RP. These high-dimensional results may seem to conflict with the experiment in Section 2.3.3, where CSARA performance degraded quickly with the addition of further, irrelevant descriptors. CSARA's strong performance here with high-dimensional sets is probably a reflection of the quality of the sets, where all variables may be at least weakly informative. In contrast, irrelevant variables are completely unrelated to activity.

Table 2.6 displays the significance levels for tests of a difference in mean HR, comparing CSARA, Cluster/RP, and Random/RP pairwise for each of the four descriptor sets. It is clear that CSARA has a significantly larger mean HR than Cluster/RP or Random/RP, and that Cluster/RP performs better than Random/RP as K increases. A paired `t`-test was used to compare CSARA with Cluster/RP, and a two-sample `t`-test for other comparisons.

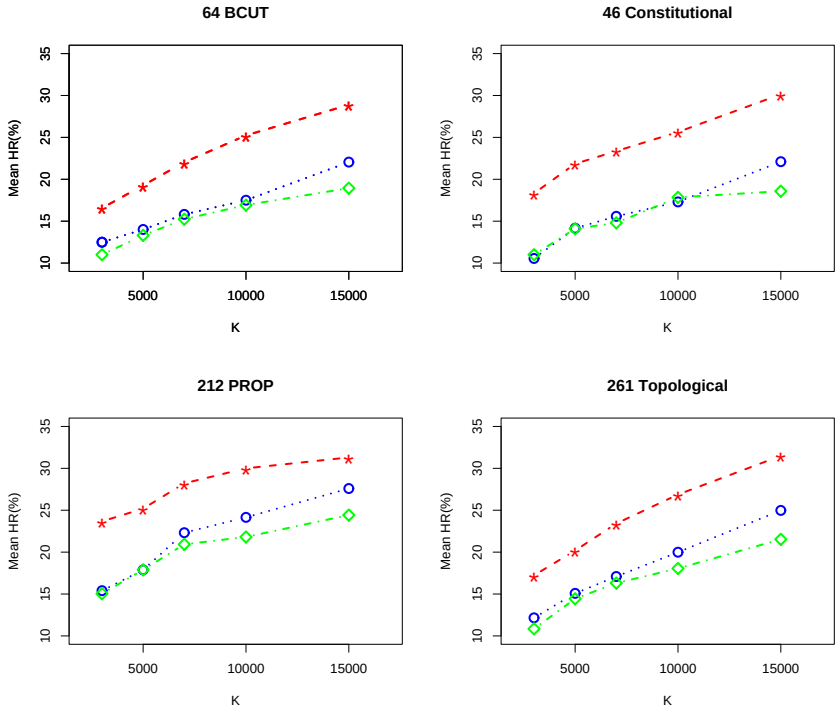

Figure 2.4: Mean hit rate versus K for CSARA ($\star\cdots\star$), Cluster/RP ($\circ\cdots\circ$), and Random/RP ($\diamond\cdots\diamond$) for the AIDS data with four high-dimensional descriptor sets.

2.5 Application of CSARA to a Continuous Assay Response

As mentioned in Section 1.2.1, growth inhibition (potency) of compounds was originally measured as a continuous response for the Yeast assay. "Active" and "inactive" labels were obtained by thresholding the response. Here, we adapt CSARA for a continuous response, and compare performance with RP methods.

Step D of the CSARA algorithm in Section 2.1.1 is adapted as follows. Each cluster is scored according to the potency of the compound randomly sampled from

	CSARA vs. Cluster/RP				CSARA vs. Random/RP				Cluster/RP vs. Random/RP			
K	BCUT	CON	PROP	TOP	BCUT	CON	PROP	TOP	BCUT	CON	PROP	TOP
3,000	***	***	***	***	***	***	***	***				
5,000	***	***	***	***	***	***	***	***				
7,000	***	***	***	***	***	***	***	***				
10,000	***	***	***	***	***	***	***	***			**	*
15,000	***	***	***	***	***	***	***	***	***	***	***	***

Table 2.6: Hypothesis tests comparing mean HR for three sampling/analysis strategies when applied to the AIDS data with four descriptor sets: 64 BCUT descriptors, 46 Constitutional descriptors, 212 Property descriptors and 261 Topological descriptors. Differences in mean HR significant at the 5%, 1%, and 0.1% levels are denoted by *, **, and ***, respectively.

it for assay. All compounds in the highest scoring cluster are chosen first for the second-round assay, then those in the second-highest scoring cluster, and so on. Building an RP *regression* tree for a continuous response is well known (Breiman et al. 1984). Comparisons are carried out with the six BCUT descriptors and $K = 3,000$.

The results are shown graphically in Figure 2.5. The average potency of the selected compounds is plotted against the number of compounds selected. A curve that is high at the left and decreases gradually would indicate good ability to identify high potency compounds.

Several observations can be made on the basis of Figure 2.5:

- Up to about 1000 compounds selected, there is substantial improvement over random selection, which would correspond to a horizontal line at a height of approximately 8%.

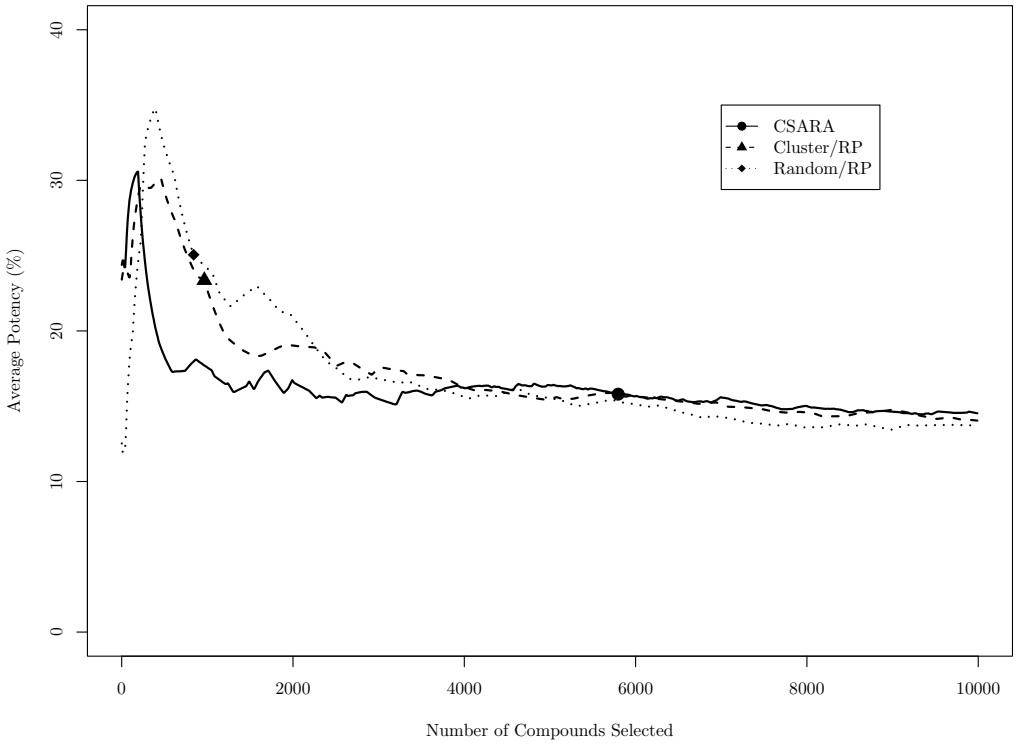

Figure 2.5: Average potency (%) of selected compounds versus the number of compounds selected for CSARA, Cluster/RP, and Random/RP, when $K = 3,000$. The symbols on the curves show where selection would stop if a predicted potency larger than 70% inhibition is required.

- Compared with the RP methods, performance of CSARA falls off faster with the number of compounds selected.
- If 70% predicted inhibition is used as a cut-off, CSARA chooses more compounds for the second screen compared with the RP strategies.

Similar results (not shown) are obtained for $K = 10,000$, except that all strategies, not surprisingly, return higher average potencies for the first 10,000 compounds selected with a larger K.

2.6 Discussion

The main aim of this chapter is exploring the properties of CSARA. Two data sets are analyzed to help evaluate the differences between CSARA and RP. The results suggest that CSARA outperforms RP models in selecting more hits. RP trees are trained to give overall good prediction, giving equal weight to both active and inactive compounds. In contrast, in its very simple analysis of the first-round training data, CSARA gives heavy weight (100%) to active compounds and no weight to inactive compounds. Thus, CSARA may be a more appropriate method for drug discovery data sets where actives are rare.

However, the largest limitation with CSARA is its instability. When there are many irrelevant descriptors, the effectiveness of CSARA decreases. In the presence of many potential irrelevant descriptors, variable selection may need to be carried out first. Here the term "irrelevant" variable refers to those that have no relationship with the biological assay and hence have a negative impact on QSAR model

training and prediction. For the AIDS assay data and the four high-dimensional descriptor sets used in the chapter, some experiments have been done to test if there is any performance improvement of CSARA, Cluster/RP, and Random/RP using the important variables identified by Partitionator from `www.goldenhelix.com`. No significant improvements were found for the three approaches. It is not surprising that the performance of Cluster/RP and Random/RP is not improved by variable selection, as trees can automatically select important variables at each split. Here we need to emphasize that the variables that are not chosen as important are not necessarily irrelevant. They seemingly do not help in the prediction of biological activity, but they are not harmful to prediction either, possibly because of correlations with other variables. This may explain why variable selection does not improve CSARA for these descriptor sets.

One potential shortcoming of CSARA is the lack of control over the number of compounds selected for a second HTS. In applications where the categorical activity is formed from an underlying continuous response, the methods in Section 2.5 may circumvent this difficulty.

The results in Section 2.5 also provide much insight into the comparisons between CSARA and the RP methods. RP trees may be effective in choosing a relatively small number of second-screen (test) compounds. In most of the experiments in this chapter, however, we matched the number of test compounds across CSARA and the RP methods, forcing RP to select a large number of compounds if CSARA does. When control over the size of the second screen is made possible for CSARA too (by working with a continuous assay measurement), RP methods

can become more competitive for smaller screens.

Chapter 3

Introduction To Mixture Discriminant Analysis

In Chapter 2, CSARA successfully separated data into different classes and outperformed tree models. An essential component of CSARA is its use of clustering to subdivide the data into groups more likely to be active. This motivates us to consider using mixture models, a model-based clustering technique, to classify data. Although mixture models are an unsupervised learning technique, they can be used in discriminative models, in which the joint distributions of descriptors are modeled as a mixture, conditioned on the response class.

Section 3.1 gives an overview of discriminant analysis approaches. Since mixture models will be used as a component of a discriminant analysis method, an introduction to mixture models is given in Section 3.2. An introduction to mixture discriminant analysis is presented in Section 3.3, and some motivation for the use of mixture discriminant analysis method in drug discovery is given in Section 3.4.

3.1 Overview of Discriminant Analysis

Discriminant Analysis refers to a variety of models designed for classification (i.e. the assignment of the data into predefined classes). In general, the number of classes is assumed to be known. Many discriminant analysis methods are probabilistic, based on the assumption that the observations in the k^{th} class are generated by a probability distribution specific to that class $f(\mathbf{x}; \boldsymbol{\Psi_k})$, also called the class-conditional distribution. Discriminant analysis models differ essentially in their assumptions about the class-conditional distribution.

If τ_k is the proportion of members of the population that are in class k, Bayes' theorem says that the posterior probability that an observation with feature vector $\mathbf{x}$ belongs to class k is

$$Pr[\text{Class k}|\mathbf{x}] = \frac{\tau_k f(\mathbf{x}; \boldsymbol{\Psi_k})}{\sum_{l=1}^{K} \tau_l f(\mathbf{x}; \boldsymbol{\Psi_l})}, \tag{3.1}$$

where K is the total number of classes in the data. Then $\mathbf{x}$ is assigned to the class with the highest posterior probability.

The most common discriminant analysis method, linear discriminant analysis (LDA), assumes that the class conditional distributions are P-variate normal with mean vectors $\mu_\mathbf{k}$ and common covariance matrix $\boldsymbol{\Sigma}$. When the covariance matrices $\boldsymbol{\Sigma_k}$'s are not assumed equal, the method is called quadratic discriminant analysis (QDA). The parameters $\mu_\mathbf{k}$ and $\boldsymbol{\Sigma_k}$ are unknown and must be estimated from a training set consisting of $(\mathbf{x_i}, y_i)$, $i = 1, \ldots, n$, where $\mathbf{x_i}$ is a vector-valued measurement and $y_i \in \{1, 2, \ldots, K\}$ is a class indicator for observation i. The parameters are generally chosen to maximize the likelihood of the training sample. This leads

to the maximum likelihood estimates

$$\hat{\mu}_{\mathbf{k}} = \mathbf{x_k} = \frac{\sum_{y_i=k} \mathbf{x_i}}{n_k}, \qquad 1 \leq k \leq K. \tag{3.2}$$

where $n_k = \sum_{i=1}^{n} I(y_i = k)$. For LDA,

$$\hat{\mathbf{\Sigma}} = \frac{\boldsymbol{S}}{n} = \frac{\sum_{k=1}^{K} \sum_{y_i=k} (\mathbf{x_i} - \bar{\mathbf{x}}_{\mathbf{k}})(\mathbf{x_i} - \bar{\mathbf{x}}_{\mathbf{k}})^T}{n}; \tag{3.3}$$

for QDA,

$$\hat{\mathbf{\Sigma}}_{\boldsymbol{k}} = \frac{\boldsymbol{S_k}}{n_k} = \frac{\sum_{y_i=k} (\mathbf{x_i} - \bar{\mathbf{x}}_{\mathbf{k}})(\mathbf{x_i} - \bar{\mathbf{x}}_{\mathbf{k}})^T}{n_k}, \qquad 1 \leq k \leq K. \tag{3.4}$$

Friedman (1989) considered linear and quadratic discriminant analysis in a small sample, high-dimensional setting. He proposed Regularized Discriminant Analysis (RDA), which employs an alternative to the usual maximum likelihood estimates for the covariance matrices. RDA specifies the value of a complexity parameter and of a shrinkage parameter to design an intermediate classifier between the linear, the quadratic, and the nearest-means classifiers. RDA performs well but does not provide easily interpretable classification rules.

Bensmail & Celeux (1996) proposed an alternative approach for discriminant analysis problem, Eigenvalue Decomposition Discriminant Analysis (EDDA). EDDA is based on the reparameterization of the covariance matrix $\mathbf{\Sigma_k}$ of class k in terms of its eigenvalue decomposition $\mathbf{\Sigma_k} = \lambda_k \boldsymbol{D_k} \boldsymbol{A_k} \boldsymbol{D_k}^T$ (Banfield & Raftery 1993). Here λ_k specifies the volume of density contours of class k; $\boldsymbol{A_k}$, the diagonal matrix of eigenvalues, specifies the shape of class k; and $\boldsymbol{D_k}$, the eigenvectors, specifies its orientation. Variations on constraints concerning volumes, shapes and orientations λ_k, $\boldsymbol{A_k}$ and $\boldsymbol{D_k}$ lead to 14 discrimination models of interest. After the class-conditional

distributions are determined, observations are assigned to the class with the largest posterior probability (3.1).

3.2 Mixture Models

Finite mixture models have wide applications in the scientific literature. They provide a mathematical approach to the statistical modeling of a wide variety of random phenomena. Mixture distributions are typically used to model data in which each observation is assumed to have arisen from one of J different groups, each group being modeled by a probability density belonging to a parametric family. Membership in the groups is not observed. Mixture models are suitable for clustering observations together into groups.

The first attempts to analyze mixture models are often attributed to Pearson (1894), who applied mixture models to data on the dimensions of crabs. Since then, mixture models have been used in a large range of applications. McLachlan & Basford (1988) highlighted the important role of mixture models in the field of cluster analysis. In the cluster analysis framework, the data is supposed to be sampled from some population described by a probability density function. This density function is characterized by a parameterized model that is a mixture of component density functions and each component density function describes one of the clusters.

In general, let $f(\mathbf{x}; \mathbf{\Psi})$ be a parametric density function with respect to some σ-finite measure and parameter space $\ominus$, which is usually a subset of some Euclidean

space. The density function of a finite mixture model is given by

$$f(\mathbf{x}; \mathbf{\Psi}) = \sum_{j=1}^{J} \pi_j f(\mathbf{x}; \mathbf{\Phi}_j) \tag{3.5}$$

where J is the number of components or the order of the model, and $\mathbf{\Psi}$ represents all the parameters in the above density function and includes $\{\pi_1, \ldots, \pi_{J-1}; \mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_J\}$. $\mathbf{\Phi}_j$ is the parameter of the j^{th} component density, and π_j is the mixing proportion of the j^{th} component density.

There are several approaches to the estimation of the parameters of mixture models. As discussed by McLachlan & Peel (2000), such approaches include graphical methods, methods of moments, minimum-distance methods, maximum likelihood estimation (MLE) and Bayesian methods. Maximum likelihood estimation is by far the most commonly used approach. Such popularity is mainly due to the advent of the Expectation-Maximization (EM) algorithm (Dempster, Laird & Rubin 1977), which is an iterative method that locally maximizes the likelihood function in an efficient way. The EM algorithm not only considerably simplifies the MLE approach to mixture parameter estimation by viewing it as an incomplete-data problem, but also gives a theoretical basis for the convergence properties of mixture problems.

Therefore, maximum likelihood estimation via the EM algorithm is the approach we consider in the following chapters.

A popular choice of component density is the normal distribution. The earlier researchers who have studied mixtures of normal distributions include Day (1969), Wolfe (1970), Marriott (1975) and Symons (1981). Mixtures of other distributions that have been considered by other researchers include exponential (Rider 1961),

Beta (Bremmer 1978), Weibull (Kao 1959) and Binomial (Blischke 1962, Rider 1962, Blischke 1964). In the following chapters, mixtures of normal distributions will be our focus.

In the next section, we will focus on discussion of mixture discriminant analysis, in which mixtures are used within each response class.

3.3 Mixture Discriminant Analysis

An alternative model-based approach to generalizing LDA and QDA is to allow the density for each class itself to be a mixture of normals, namely

$$f(\mathbf{x}; \boldsymbol{\Psi_k}) = \sum_{j=1}^{J_k} \pi_{jk} MVN(\mathbf{x}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}). \tag{3.6}$$

Here k indexes class and j the mixture component within class. $\boldsymbol{\Psi_k}$ represents all the parameters within class k, i.e. $\boldsymbol{\Psi_k} = \{\pi_{1k}, \ldots, \pi_{J_k-1k}; \boldsymbol{\mu_{1k}}, \ldots, \boldsymbol{\mu_{J_k k}}; \boldsymbol{\Sigma_{1k}}, \ldots, \boldsymbol{\Sigma_{J_k k}}\}$. Here, and throughout the remainder of the thesis, we denote a multivariate normal density with $MVN(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. A univariate normal density will be denoted $N(x; \mu, \sigma)$. The idea of mixture models has been suggested a number of times in the literature (McLachlan 1992), and is the basis of Mixture Discriminant Analysis or MDA (Hastie & Tibshirani 1996). In developing MDA, Hastie & Tibshirani (1996) made two assumptions: (i) that all of the component covariance matrices are the same, i.e. $\boldsymbol{\Sigma_{jk}} = \boldsymbol{\Sigma}$ for each j, k; and (ii) that the number of mixture components is known in advance for each class. Hastie & Tibshirani (1996) also proposed several extensions of the model under these assumptions. Moreover, Fraley

& Raftery (2002) extended MDA by relaxing assumptions (i) and (ii) and applying model-based clustering to the members of each class in the training set. This would allow the component covariance matrices to vary, both within and between classes. The data would then determine which parametrization of the covariance matrix and which number of mixture components is best suited to each class. This generalization of MDA is refered as Model-based Clustering Discriminant Analysis (MclustDA).

The basic idea of the model-based discriminant analysis methods described here is to allow more flexibility than the traditional methods, LDA and QDA. Also mixture-based MDA and MclustDA further improve on EDDA by expanding the discriminant model from a single Gaussian component to a mixture.

However, none of the above discriminant methods considers exploring subsets of predictors, which is becoming critical for higher-dimensional drug discovery data due to the subset-governed activities. In our new form of mixture discriminant analysis, $\boldsymbol{\mu_{jk}}$ and $\boldsymbol{\Sigma_{jk}}$ share some common parameters, called global parameters in our model. Each component is dominated by one element of the descriptor vector, so the structure of the new mixture discriminant analysis model is designed to explore subsets of descriptor space.

3.4 Motivation of Application of Mixture Model in Drug Discovery

There are numerous papers on the application of mixture models, but there are fewer applications of mixture models in drug discovery. There are several considerations motivating us to use mixture models to model drug discovery data.

These considerations come from the HTS data sets themselves. Although we have mentioned these ideas in Section 1.3, we develop them further here:

(1) Because of the complicated relationship between descriptors and biological activity of compounds, it is difficult to use a single mathematical model to capture the characteristics of the entire data set. As we have discovered in the first part of the thesis, the active compounds can usually be divided into several clusters. The active compounds do have the same effect on the drug target, but across clusters, they can have very different descriptors leading into activities. Special models, such as mixture models are needed to model several different mechanisms simultaneously. This is the multiple mechanism problem.

(2) The biological activity is usually governed by a small number of descriptors. Due to the flexibility of mixture models, making some modifications on mixture models would allow us to explore subsets of descriptors.

(3) There are many descriptors, which are often highly correlated. Again, CSARA and tree models can not take into account the covariance of descriptors, but

mixture models can. For example, the covariance matrix of a multivariate normal distribution can describe such properties of descriptors.

In the thesis, we focus on the first and second considerations. Chapter 4 presents our specially designed mixture discriminant analysis model, Constrained Mixture Discriminant Analysis (CMDA). The CMDA first order model or CMDA1 is discussed in detail. A frequently-occurring problem arises, in which the EM algorithm can produce "degenerate" estimates with some variances equal to zero. This occurs because the likelihood for a mixture model with unknown scale parameters is unbounded. Therefore, a Multi-step EM algorithm is designed to solve this problem in Chapter 4. In Chapter 5, penalized maximum likelihood estimation approach to solve the degeneracy problem is suggested and discussed. A consistency proof of the penalized maximum likelihood estimate (PMLE) for the CMDA1 model with a two-dimensional descriptor space is provided in Chapter 5.

Chapter 4

Constrained Mixture Discriminant Analysis

4.1 Introduction

Statistical learning in drug discovery seeks a good classifier that separates chemical compounds into active and inactive classes. However, the characteristics of drug data imply many challenges for structure modeling and identification of active compounds (Section 1.3 and Section 3.4). Due to the characteristics of drug discovery data sets, we develop the Constrained Mixture Discriminant Analysis (CMDA) model, which is designed to catch multiple mechanisms that lead to activity, explore the subsets of descriptors and be easily interpreted (e.g. identify important descriptors).

The approach to classification taken here is to model the within-class densities of the predictors $f(\mathbf{x}|\text{Class k})$ by constrained mixture models. Then the class pos-

terior probabilities $f(\text{Class k}|\mathbf{x})$ can be obtained via Bayes' theorem as described in Section 3.1.

Before the model is given, we review some notation introduced earlier in Chapter 3:

- y is the response variable taking K categorical levels, which is biological activity in drug discovery. Usually, in drug discovery, $K = 2$, i.e. the active and inactive classes.
- $\mathbf{x}$ is a vector of descriptors from an observation, which is assumed to come from a P-dimensional real-valued sample space, i.e. $\mathbf{x} = (x_1, \ldots, x_P) \in \Re^P$.
- $k = 1, \ldots, K$ indexes the K classes.
- $j = 1, \ldots, J_k$ indexes components in each class. J_k is the total number of components in k^{th} class.
- $i = 1, \ldots, n$ indexes observations in the sample.

The CMDA model is based on the belief that the influence of descriptor vector $\mathbf{x}$ on biological activity y is through low-dimensional subspaces (Lam 2001). We shall express the class-conditional density of descriptor vector $\mathbf{x}$ as an additive model including low-dimensional functions. This is designed to explore the subspaces of descriptors and identify the multiple mechanisms that may cause activity.

A general form of mixture models for the density function of class k is given by

$$f(\mathbf{x}; \boldsymbol{\Psi_k}) = \sum_{j=1}^{J_k} \pi_{jk} f(\mathbf{x}; \boldsymbol{\Phi_{jk}}), \tag{4.1}$$

where π_{jk}'s are the mixing proportions of components in class k, and satisfy $\sum_{j=1}^{J_k} \pi_{jk} = 1$.

In the CMDA model, the multivariate density of any component $f(\mathbf{x}; \boldsymbol{\Phi_{jk}})$ will be composed of products of univariate or bivariate density functions. Types of $f(\mathbf{x}; \boldsymbol{\Phi_{jk}})$ that could be considered, when $P = 4$ and the normal densities are used, include:

1. $f(\mathbf{x}; \boldsymbol{\Phi_{jk}}) = N(x_1; \mu_{1j}, \sigma_{1j})N(x_2; \mu_{2j}, \sigma_{2j})N(x_3; \mu_{3j}, \sigma_{3j})N(x_4; \mu_{4j}, \sigma_{4j})$. The components of $\mathbf{x}$ (x_l's) are independent and unrelated to the class label k.

2. $f(\mathbf{x}; \boldsymbol{\Phi_{jk}}) = N(x_j; \mu_{jk}, \sigma_{jk}) \prod_{m \neq j}^{4} N(x_m; \mu_m, \sigma_m)$, i.e. one element of the descriptor vector is conditionally independent of other elements given the class label k and the mixture component label j. Also $N(x_j; \mu_{jk}, \sigma_{jk})$ is a class specific density with parameters depending on both j and k. There is a connection between variables and the component in this type of component density, i.e. component $f(\mathbf{x}; \boldsymbol{\Phi_{jk}})$ is determined by variable x_j via the function $N(x_j; \mu_{jk}, \sigma_{jk})$.

3. $f(\mathbf{x}; \boldsymbol{\Phi_{jk}}) = MVN(x_l, x_{l'}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}) \prod_{m \neq l \& l'} N(x_m; \mu_m, \sigma_m)$, which means there are two elements of the descriptor vector that have a jointly dependent relationship given the class label k. Here, j corresponds to the pair $(l, l^{'})$.

4. $f(\mathbf{x}; \boldsymbol{\Phi_{jk}})$'s can be a combination of above forms, for example

$$f(\mathbf{x}; \boldsymbol{\Phi_{jk}}) = N(x_1; \mu_1, \sigma_1)N(x_2; \mu_{jk}, \sigma_{jk})MVN(x_3, x_4; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}) \ldots \quad (4.2)$$

Our primary focus will be on model of form 2. Form 3 will also be considered. Although we use bivariate distributions to represent joint dependence in Form 3, in

general, these need not be bivariate. More explicit parameterizations will be given in the next section.

We first consider the CMDA First Order Model, or the CMDA1 model,

$$\begin{aligned} f(\mathbf{x};\boldsymbol{\Psi_k}) = \pi_{1k}N(x_1;\mu_{1k},\sigma_{1k})\prod_{j=2}^{P}N(x_j;\mu_j,\sigma_j) \\ + \pi_{2k}N(x_2;\mu_{2k},\sigma_{2k})\prod_{j\neq 2}N(x_j;\mu_j,\sigma_j) + \ldots \\ + \pi_{Pk}N(x_P;\mu_{Pk},\sigma_{Pk})\prod_{j\neq P}N(x_j;\mu_j,\sigma_j). \end{aligned}$$

In the following section, a detailed discussion of the CMDA1 model is given.

4.2 The CMDA First Order Model (CMDA1)

For a general case, including the parameters in the density function, the CMDA1 mixture density for class k is

$$f(\mathbf{x};\boldsymbol{\Psi_k},\boldsymbol{\Psi_G}) = \sum_{j=1}^{P}\pi_{jk}h(x_j;\bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}})\prod_{l\neq j}h(x_l;\bar{\boldsymbol{\Phi}}_{\boldsymbol{l}}), \tag{4.3}$$

where h is some univariate density function with parameters $\bar{\boldsymbol{\Phi}}$ and

$$\boldsymbol{\Psi_k} = (\pi_{1k},\ldots,\pi_{(P-1)k},\bar{\boldsymbol{\Phi}}_{\mathbf{1k}},\ldots,\bar{\boldsymbol{\Phi}}_{\boldsymbol{Pk}})^T$$

represents all the unknown parameters specific to class k. The model also has global parameters $\boldsymbol{\Psi_G} = (\bar{\boldsymbol{\Phi}}_{\mathbf{1}},\ldots,\bar{\boldsymbol{\Phi}}_{\boldsymbol{P}})^T$, which are used within all classes. For later notational convenience, we use a "G" subscript on $\boldsymbol{\Psi}$ to denote a collection of "global" parameters, and a "k" subscript on $\boldsymbol{\Psi}$ to index parameters specific

to class k. Here and throughout the thesis, we use a single subscript (e.g. $\bar{\boldsymbol{\Phi}}_{\boldsymbol{l}}$) to denote "global" parameters and double subscripts (e.g. $\bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}}$) to denote class-specific parameters. Then all parameters are denoted as $\boldsymbol{\Psi} = (\boldsymbol{\Psi_1}, \ldots, \boldsymbol{\Psi_K}, \boldsymbol{\Psi_G})^T$.

There are some interpretations for the CMDA1 model:

- Each component density is a product of P univariate density functions, which suggests the CMDA1 model is based on an assumption that all the descriptors are independent given a class and a component. These components could be thought of as corresponding to specific mechanisms.

- Since the CMDA1 model explores all the 1-dimensional subsets of the descriptors, the number of components in each class equals P, the dimension of $\mathbf{x}$. That is, in the general form of the mixture model (4.1), $J_k = P$ for $k = 1, \ldots, K$. Each component is primarily identified by only one element of $\mathbf{x}$, whose distribution depends on class labels, via the term $h(x_j; \bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}})$.

- There are two parts in each component density function: "$h(x_j; \bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}})$", the class specific part and "$h(x_l; \bar{\boldsymbol{\Phi}}_{\boldsymbol{l}})$", the global part. In (4.3), $h(x_j; \bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}})$ varies across both mixture components and classes while the $h(x_l; \bar{\boldsymbol{\Phi}}_{\boldsymbol{l}})$ terms in the product remain the same across classes. The concept of global parameters comes from the reality that the active compounds are rare in typical drug data. It is hard to accurately estimate the parameters for the density function of the rare class due to the small samples of active compounds. Using global densities allows the estimation for the rare class to borrow strength across classes.

- There are fewer parameters to be estimated for the CMDA1 model compared to an unconstrained case, in which each component has distinct parameter values. For instance, if $h(\cdot)$ in (4.3) are assumed to be normal density distributions, then the CMDA1 model is:

$$f(\mathbf{x}; \boldsymbol{\Psi_k}, \boldsymbol{\Psi_G}) = \sum_{j=1}^{P} \pi_{jk} N(x_j; \mu_{jk}, \sigma_{jk}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l). \tag{4.4}$$

  There are in total $(3P-1) \times K + 2P$ parameters (there are $(P-1) \times K$ π's, $P \times K$ class specific μ's, $P \times K$ class specific σ's, P global μ's and P global σ's). In comparison, the traditional mixture model has $(2P^2 + P - 1) \times K$ parameters to be estimated under the same independence assumption. The traditional mixture model with the same independence assumption is

$$f(\mathbf{x}; \boldsymbol{\Psi_k}) = \sum_{j=1}^{P} \pi_{jk} MVN(\mathbf{x}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}), \tag{4.5}$$

  There are no constraints on the mean vectors ($\boldsymbol{\mu_{jk}}$'s), and the covariance matrices ($\boldsymbol{\Sigma_{jk}}$'s) are assumed to be diagonal.

Figure 4.1 illustrates the basic idea behind the CMDA1 model. The two-dimensional data are simulated from a CMDA1 model. The blue and red points represent two different classes. Consider the two clusters (one red and one blue) in the top-left corner of the plot: these two clusters share a common mean (μ_2) along the x_2 direction, and different means (μ_{11}, μ_{21}) along the x_1 direction. Hence these two clusters can be distinguished using only the descriptor x_1. The parameter μ_2 is called a global parameter, while μ_{11} and μ_{21} are local parameters. The same interpretation is applied to the other two clusters, with the global parameters in

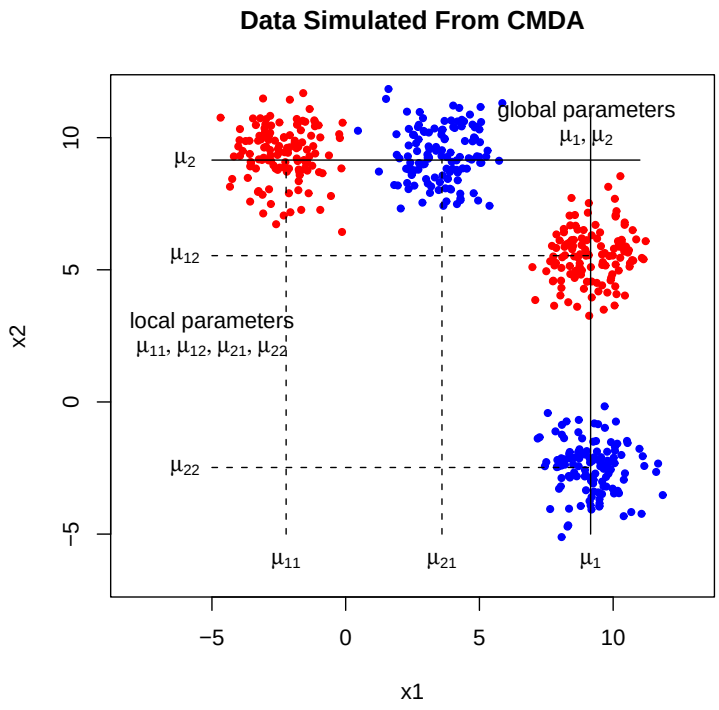

Figure 4.1: Example of the CMDA1 model. Red and blue points indicate class membership.

dimension x_1. For illustrative purpose, this plot focuses on the location parameters of the CMDA1 model. As (4.4) indicates, the variances are similarly parameterized.

4.3 The Expectation-Maximization (EM) Algorithm and Mixture Models

The asymptotic efficiency of maximum likelihood estimation makes it one of the most commonly used estimation approaches (Lindsay 1995). Maximum likelihood

estimation became popular especially after Dempster et al. (1977) introduced the EM algorithm, which can solve difficult MLE problems. Since its inception, the EM algorithm has attracted considerable attention and has been the subject of much research. The EM algorithm greatly stimulated interest in the use of finite mixture distributions to model heterogenerous data. This is because the fitting of mixture models by maximum likelihood is simplified considerably by the EM algorithm.

Since the EM algorithm will be used extensively in estimating the CMDA1 model, we review it for the simpler case of a $g-$component mixture, i.e. $f(\mathbf{x};\mathbf{\Psi}) = \sum_{j=1}^{g}\pi_j f(\mathbf{x};\mathbf{\Phi}_j)$ where $\mathbf{\Psi} = (\pi_1,\ldots,\pi_{g-1};\mathbf{\Phi}_1,\ldots,\mathbf{\Phi}_g)$. The CMDA1 model will be considered later in Section 4.4.

In the EM framework, the observed data $\mathbf{x} = (\mathbf{x}_1^T,\ldots,\mathbf{x}_n^T)^T$ is viewed as being incomplete. We introduce an associated component-label vector $\mathbf{z} = (\mathbf{z}_1^T,\ldots,\mathbf{z}_n^T)^T$, which is assumed unknown. Each $\mathbf{x}_i^T$ is conceptualized as having arisen from one of the components of the mixture model. If the mixture model has g components, $\mathbf{z}_i$ is a $g-$dimensional vector with $z_{ij} = (\mathbf{z}_i)_j = 1$ or 0, according to whether $\mathbf{x}_i$ did or did not arise from the j^{th} component of the mixture $(i = 1,\ldots,n; j = 1,\ldots,g)$. Exactly one element of $\mathbf{z_i}$ will be 1. The complete-data is defined as

$$\mathbf{x}_c = (\mathbf{x}^T,\mathbf{z}^T)^T, \tag{4.6}$$

where $\mathbf{z} = (\mathbf{z}_1^T,\ldots,\mathbf{z}_n^T)^T$.

Then the complete-data log likelihood for $\mathbf{\Psi}$, $l_c(\mathbf{\Psi})$, is given by

$$l_c(\mathbf{\Psi}) = \sum_{j=1}^{g}\sum_{i=1}^{n} z_{ij}\{\log\pi_j + \log f(\mathbf{x}_i;\mathbf{\Phi}_j)\}. \tag{4.7}$$

4.3.1 E-Step

The EM algorithm is applied to this problem by treating the z_{ij} as missing data. It proceeds iteratively in two steps, E (for expectation) and M (for maximization). The missing data (z_{ij}) are handled by the E-step, which takes the conditional expectation of the complete-data log likelihood, $l_c(\mathbf{\Psi})$ in (4.7), given the observed data $\mathbf{x}$, using the current estimate for $\mathbf{\Psi}$. Let $\hat{\mathbf{\Psi}}^{(m)}$ be the value of $\mathbf{\Psi}$ after the m^{th} EM iteration. Then on the $(m+1)^{th}$ iteration, the E-step requires the computation of the conditional expectation of $l_c(\mathbf{\Psi})$ given $\mathbf{x}$, using $\hat{\mathbf{\Psi}}^{(m)}$ for $\mathbf{\Psi}$, which can be written as

$$Q(\mathbf{\Psi}; \hat{\mathbf{\Psi}}^{(m)}) = E\{l_c(\mathbf{\Psi})|\mathbf{x}, \hat{\mathbf{\Psi}}^{(m)}\}. \tag{4.8}$$

In the expectation, we take $\mathbf{\Psi} = \hat{\mathbf{\Psi}}^{(m)}$. The expectation is with respect to the unobserved z_{ij}.

As the complete-data log likelihood, $l_c(\mathbf{\Psi})$, is linear in the unobservable data z_{ij}, the E-step (on the $(m+1)^{th}$ iteration) simply requires the calculation of the current conditional expectation of Z_{ij} given the observed data $\mathbf{x}$, where Z_{ij} is the random variable corresponding to z_{ij}. Now

$$\begin{aligned} E(Z_{ij}|\mathbf{x}, \hat{\mathbf{\Psi}}^{(m)}) &= p\{Z_{ij} = 1|\mathbf{x}, \hat{\mathbf{\Psi}}^{(m)}\} \\ &\equiv \hat{z}_{ij}, \end{aligned} \tag{4.9}$$

where

$$\hat{z}_{ij} = \hat{\pi}_j^{(m)} f(\mathbf{x}_i; \hat{\mathbf{\Phi}}_j^{(m)}) / \sum_{l=1}^{g} \hat{\pi}_l^{(m)} f(\mathbf{x}_i; \hat{\mathbf{\Phi}}_l^{(m)}), \tag{4.10}$$

for $i = 1, \ldots, n; j = 1, \ldots, g$. According to (4.10), $\hat{z}_{ij}$ can take values between 0 and 1. Using (4.10), (4.8) can be written as

$$Q(\boldsymbol{\Psi}; \hat{\boldsymbol{\Psi}}^{(m)}) = \sum_{j=1}^{g} \sum_{i=1}^{n} \hat{z}_{ij}^{(m)} \{\log \pi_j + \log f(\mathbf{x}_i; \boldsymbol{\Phi}_j)\}. \tag{4.11}$$

4.3.2 M-Step

The M-step on the $(m+1)^{th}$ iteration requires the global maximization of $Q(\boldsymbol{\Psi}; \hat{\boldsymbol{\Psi}}^{(m)})$ with respect to $\boldsymbol{\Psi}$ over the parameter space $\ominus$ to give the updated estimate $\hat{\boldsymbol{\Psi}}^{(m+1)}$. That is, we seek the estimates of π's and $\boldsymbol{\Phi}$'s using $\hat{z}_{ij}^{(m)}$ in (4.11). One nice feature of the EM algorithm is that for many common component densities f_j, the solution in the M-step often exists in closed form. Since the solution to the M-step depends on the form of density chosen, we delay further details until the next section.

4.4 EM for the CMDA1 Model

The EM algorithm can be generalized to mixture discriminant analysis problems. In this section, we derive the EM algorithm for the CMDA1 model using univariate normal densities to construct the component densities, as in (4.4). The derivation is similar to Section 4.3, but it will be more complicated because of the form of the CMDA1 model.

Knowing that observation i is in the k^{th} class, $\mathbf{z}_i = (z_{i1k}, \ldots, z_{iPk})^T$ is a P-dimensional vector such that

$$z_{ijk} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ belongs to the } j^{th} \text{ component of the } k^{th} \text{ class,} \\ 0 & \text{otherwise.} \end{cases}$$

Here $\sum_j z_{ijk} = 1$ for an observation i in a given class k. If $y_i = k$, we assume $z_{ijl} = 0$ for all $l \neq k$. The observations $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T)^T$ are called incomplete data, while $(\mathbf{x}_i^T, \mathbf{z}_i^T)^T$ with $i = 1, \ldots, n$ are called complete data (see Section 4.3).

Hence the log likelihood for the incomplete data is

$$l_n(\mathbf{\Psi}) = \sum_{i=1}^{n} \log f(\mathbf{x}_i; \mathbf{\Psi}|y_i), \tag{4.12}$$

where y_i is the class label for observation i and f is defined in (4.3). As before, we refer to the collection of all model parameters as $\mathbf{\Psi}$. That is, in the CMDA1 model, $\mathbf{\Psi} = \{\mathbf{\Psi_1}, \ldots, \mathbf{\Psi_k}, \mathbf{\Psi_G}\}$. For the completed data, the log likelihood is

$$l_c(\mathbf{\Psi}) = \sum_{k=1}^{K} \sum_{j=1}^{P} \sum_{i \in C_k} z_{ijk} \{\log \pi_{jk} + \log h(x_{ij}; \bar{\mathbf{\Phi}}_{jk}) + \sum_{l \neq j} h(x_{il}; \bar{\mathbf{\Phi}}_l)\}, \tag{4.13}$$

where x_{ij} is the j^{th} element of the vector $\mathbf{x_i}$, $\mathbf{x_i} = \{x_{i1}, \ldots, x_{iP}\}$. The notation $\sum_{i \in C_k}$ means summing over all observations belonging to the k^{th} class.

In the derivations below and later in applications (Section 4.7.1), we assume that the $h(\cdot)$ are univariate Gaussian densities. In the EM algorithm at iteration a, we need to take the expectation of (4.13) given the observations and the current estimates of parameters. That is, we need

$$\begin{aligned} Q(\mathbf{\Psi}; \hat{\mathbf{\Psi}}^{(a)}) \quad &= E\Bigg[\textstyle\sum_{k=1}^{K} \sum_{j=1}^{P} \sum_{i \in C_k} z_{ijk} \{\log \pi_{jk} + \log N(x_{ij}; \mu_{jk}, \sigma_{jk}) \\ &\qquad + \textstyle\sum_{l \neq j} N(x_{il}; \mu_l, \sigma_l)\} | \mathbf{x}, \hat{\mathbf{\Psi}}^{(a)}\Bigg]. \end{aligned} \tag{4.14}$$

In the **E-step**, we calculate the expectation of z_{ijk}, assuming $\mathbf{\Psi} = \hat{\mathbf{\Psi}}^{(a)}$:

$$\hat{z}_{ijk}^{(a)} = \frac{\hat{\pi}_{jk}^{(a)} N(x_{ij}; \hat{\mu}_{jk}^{(a)}, \hat{\sigma}_{jk}^{(a)}) \prod_{l \neq j} N(x_{il}; \hat{\mu}_l^{(a)}, \hat{\sigma}_l^{(a)})}{\sum_{m=1}^{P} \hat{\pi}_{mk}^{(a)} N(x_{im}; \hat{\mu}_{mk}^{(a)}, \hat{\sigma}_{mk}^{(a)}) \prod_{l \neq m} N(x_{il}; \hat{\mu}_l^{(a)}, \hat{\sigma}_l^{(a)})}. \tag{4.15}$$

Here, $\hat{z}_{ijk}$ is the posterior probability that the i^{th} observation belongs to the j^{th} component of the k^{th} class. Then

$$\begin{aligned} Q(\boldsymbol{\Psi}; \hat{\boldsymbol{\Psi}}^{(a)}) &= \sum_{k=1}^{K} \sum_{j=1}^{P} \sum_{i \in C_k} \hat{z}_{ijk}^{(a)} \{\log \pi_{jk} + \log N(x_{ij}; \mu_{jk}, \sigma_{jk}) \\ &\quad + \sum_{l \neq j} N(x_{il}; \mu_l, \sigma_l)\}. \end{aligned} \tag{4.16}$$

and the **M-step** seeks to maximize Q with respect to $\boldsymbol{\Psi}$ for fixed z_{ijk}. Differentiating (4.16) with respect to the parameters, and equating these derivatives to zero yields a system of equations that can easily be solved, giving:

$$\hat{\pi}_{jk}^{(a+1)} = \frac{\sum_{i \in C_k} \hat{z}_{ijk}^{(a)}}{\sum_{j'=1}^{p} \sum_{i \in C_k} \hat{z}_{ij'k}^{(a)}}, \tag{4.17}$$

$$\hat{\mu}_{jk}^{(a+1)} = \frac{\sum_{i \in C_k} \hat{z}_{ijk}^{(a)} x_{ij}}{\sum_{i \in C_k} \hat{z}_{ijk}^{(a)}}, \tag{4.18}$$

$$\hat{\sigma}_{jk}^{2\,(a+1)} = \frac{\sum_{i \in C_k} \hat{z}_{ijk}^{(a)} (x_{ij} - \hat{\mu}_{jk}^{(a)})^2}{\sum_{i \in C_k} \hat{z}_{ijk}^{(a)}}, \tag{4.19}$$

$$\hat{\mu}_{l}^{(a+1)} = \frac{\sum_{k=1}^{K} \sum_{j=1 \& j \neq l}^{P} \sum_{i \in C_k} \hat{z}_{ijk}^{(a)} x_{il}}{\sum_{k=1}^{K} \sum_{j=1 \& j \neq l}^{P} \sum_{i \in C_k} \hat{z}_{ijk}^{(a)}}, \tag{4.20}$$

$$\hat{\sigma}_{l}^{2(a+1)} = \frac{\sum_{k=1}^{K} \sum_{j=1 \& j \neq l}^{P} \sum_{i \in C_k} \hat{z}_{ijk}^{(a)} (x_{il} - \hat{\mu}_{l}^{(a)})^2}{\sum_{k=1}^{K} \sum_{j=1 \& j \neq l}^{P} \sum_{i \in C_k} \hat{z}_{ijk}^{(a)}}. \tag{4.21}$$

Finally, by plugging in the parameter estimates, the estimates of posterior class probabilities are (by Bayes' theorem)

$$\hat{P}(y = k | X = \mathbf{x}) \propto \tau_k \sum_{j=1}^{P} \pi_{jk} N(x_j; \hat{\mu}_{jk}, \hat{\sigma}_{jk}) \prod_{l \neq j} N(x_l; \hat{\mu}_l, \hat{\sigma}_l) \tag{4.22}$$

$$\sum_{k=1}^{K} \hat{P}(y = k | X = \mathbf{x}) = 1. \tag{4.23}$$

where τ_k is a prior probability for the class k.

4.5 Application Issues of the EM Algorithm

In this section, we identify two technical problems with using the EM algorithm to estimate the CMDA1 model. These problems will motivate the multi-step EM algorithm proposed in Section 4.6.

4.5.1 Degeneracy

When applying the EM algorithm to the CMDA1 model, we notice that the EM algorithm can converge to degenerate solutions. That is, the estimates of some variances are zero or very close to zero. Usually this occurs because a mixture component has only one observation associated with it. Figure 4.2 shows one example. The data are plotted in (a). The highlighted green point in (b) is in a cluster with only itself at the last iteration of the algorithm. As the EM algorithm proceeds, $\hat{\sigma}_{12} \to 0$ and $\hat{\mu}_{12}$ approaches the observed value of x for the highlighted point. As $\hat{\sigma}_{12} \to 0$, the log likelihood goes to infinity. The reason for this problem is that the MLE is not well defined in mixture models as the likelihood function of mixture models is unbounded for any given sample size.

To see how such a degenerate solution can occur, consider a mixture of g univariate $N(\mu_j, \sigma_j)$ densities. The log likelihood function is given by

$$\begin{aligned} l_n(\mathbf{\Psi}) &= \sum_{i=1}^{n} \log f(x_i; \mathbf{\Psi}) \\ &= \sum_{i=1}^{n} \log\{\sum_{j=1}^{g} \frac{\pi_j}{\sigma_j} \phi(\frac{x_i - \mu_j}{\sigma_j})\}, \end{aligned}$$

where $\mathbf{\Psi} = \{\pi_1, \ldots, \pi_{g-1}; \mu_1, \ldots, \mu_g; \sigma_1, \ldots, \sigma_g\}$, and $\phi(\cdot)$ is a standard normal

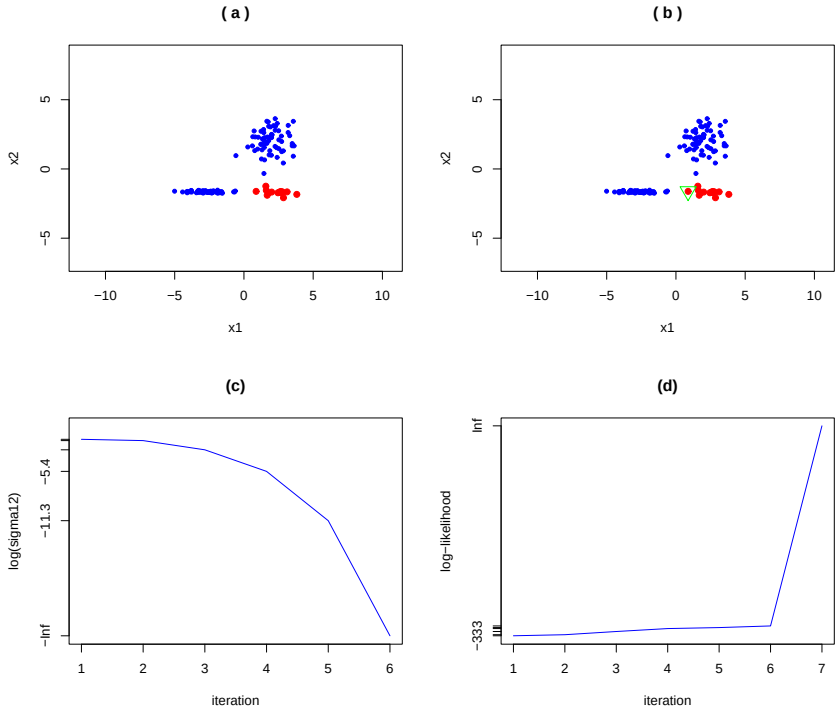

Figure 4.2: The EM algorithm converges to a degenerate solution. (a): the data set; (b): one observation x^* with the green triangle is the one that causes the degenerate solution. (c): the parameter estimate of $\log(\sigma_{12})$ decreases to $-\infty$; (d): the corresponding log likelihood diverges toward infinity. The initial values of the parameters in this example are selected by K-means. The log likelihood at iteration 7 is ∞, $\hat{\sigma}_{12}^{(6)} = 0$, and $\hat{\mu}_{12}^{(6)} = x^*$.

distribution. By letting $\mu_1 = x_i$ and $\sigma_1 \to 0$ with other parameters fixed, it is easily seen that $l_n(\mathbf{\Psi}) \to \infty$.

The problem of degenerate solutions can occur quite often. For example, later in Section 4.7.2, we show a simulated example in which 200 realizations of a data set are generated, and EM leads to a degenerate solution every time.

4.5.2 Starting Values

In mixture modeling, it is well-known that the choice of good starting values for parameters is very important in the EM algorithm (McLachlan & Krishnan 1997). Good starting values can lead the EM algorithm to converge to good local optima. Starting values can be chosen for either $\mathbf{\Psi}$ or z_{ijk}. In the thesis, the K-means algorithm is used to choose starting values for the component labels, $\hat{z}_{ijk}^{(0)}$'s. As different starting values can give very different results, we use multiple starting values in our Multi-step EM algorithm. This will help the EM algorithm identify good parameter estimates. A detailed discussion of the choice of starting values for the Multi-step EM algorithm is given in Section 4.6, and assessment of estimate quality is given in Section 4.7.

4.6 Multi-step EM Algorithm

We develop a special Multi-step EM algorithm, which is designed to improve the chances of finding good local optima and avoiding degenerate estimates. The Multi-step EM algorithm has three important features:

(1) Multiple trials for K-means are used to identify good starting points.

(2) During an intermediate stage of the algorithm, variances are enlarged to prevent degenerate solutions.

(3) The algorithm first optimizes the location parameters while holding the variance parameters fixed. Then the variances and location parameters are simultaneously optimized. Section 4.5 provides motivation for this by noting that a poor μ estimate (e.g. $\mu = x_i$ for some i) can lead to degeneracy in σ.

Because the Multi-step EM algorithm is sensitive to the starting points, (1) is used to identify good starting values. Features (2) and (3) are combined to avoid the degeneracy problem. The pseudo code of the Multi-step EM algorithm is listed below:

Step 1 **Goal: get initial values of $\hat{\pi}_{jk}^{(0)}$, $\hat{\mu}_{jk}^{(0)}$, $\hat{\mu}_{l}^{(0)}$, $\hat{\sigma}_{jk}^{2\,(0)}$ and $\hat{\sigma}_{l}^{2(0)}$**

- For t=1 to `trial`
  - (a) Run the K-means algorithm to obtain the cluster labels $(1, \ldots, P)$ for each observation given class k;
  - (b) Match $\hat{z}_{ijk}^{(0)}$ and the cluster labels obtained from (a);
  - (c) Calculate $\hat{\pi}_{jk}^{(0)}$, $\hat{\mu}_{jk}^{(0)}$, $\hat{\mu}_{l}^{(0)}$, $\hat{\sigma}_{jk}^{2\,(0)}$ and $\hat{\sigma}_{l}^{2(0)}$ in (4.17)-(4.21);
  - (d) Calculate incomplete log likelihood (4.12) $log.value^t$ given $\hat{\pi}_{jk}^{(0)}$, $\hat{\mu}_{jk}^{(0)}$, $\hat{\mu}_{l}^{(0)}$, $\hat{\sigma}_{jk}^{2\,(0)}$ and $\hat{\sigma}_{l}^{2(0)}$;
- Identify the estimates giving the best $log.value^t$, $t^* = \arg\max_t log.value^t$ as the initial values for Step 2;
- Let $\hat{\sigma}_{jk}^{2\,(0)} \leftarrow \hat{\sigma}_{jk}^{2\,(0)} \times$ `multiplier` and $\hat{\sigma}_{l}^{2(0)} \leftarrow \hat{\sigma}_{l}^{2(0)} \times$ `multiplier`.

Step 2 **Goal: find the best $\hat{\mu}_{jk}$, $\hat{\mu}_{l}$ while holding $\hat{\sigma}_{jk}^{2\,(0)}$ and $\hat{\sigma}_{l}^{2(0)}$ fixed**

- Repeat, for $m = 0, 1, 2...$,
  - (a) Calculate $\hat{z}_{ijk}^{(m+1)}$ in (4.15) given $\hat{\pi}_{jk}^{(m)}$, $\hat{\mu}_{jk}^{(m)}$, $\hat{\mu}_{l}^{(m)}$, $\hat{\sigma}_{jk}^{2\,(0)}$ and $\hat{\sigma}_{l}^{2(0)}$;

(b) Calculate $\hat{\pi}_{jk}^{(m+1)}$, $\hat{\mu}_{jk}^{(m+1)}$, $\hat{\mu}_{l}^{(m+1)}$ in (4.17), (4.18) and (4.20);

(c) Calculate $log.value^{(m+1)}$ using $\hat{\pi}_{jk}^{(m+1)}$, $\hat{\mu}_{jk}^{(m+1)}$, $\hat{\mu}_{l}^{(m+1)}$, $\hat{\sigma}_{jk}^{2}{}^{(0)}$ and $\hat{\sigma}_{l}^{2(0)}$;

(d) If $log.value^{(m+1)}$ is finite and $\frac{log.value^{(m+1)} - log.value^{(m)}}{log.value^{(m)}} < 0.0001$, stop;
If $log.value^{(m+1)}$ is infinite, stop, and return "The Algorithm Converged to A Degenerate Solution";
otherwise go to (a);

- The final set of $\hat{\pi}_{jk}^{(m^*)}$, $\hat{\mu}_{jk}^{(m^*)}$, $\hat{\mu}_{l}^{(m^*)}$, $\hat{\sigma}_{jk}^{2}{}^{(0)}$ and $\hat{\sigma}_{l}^{2(0)}$ is identified as the initial values for Step 3.

Step 3 **Goal: find the best local optima of $\hat{\mu}_{jk}$, $\hat{\mu}_{l}$, $\hat{\sigma}_{jk}^{2}$ and $\hat{\sigma}_{l}^{2}$**

- Repeat, for $m = 0, 1, 2...$,

(a) Calculate $\hat{z}_{ijk}^{(m+1)}$ in (4.15) given $\hat{\pi}_{jk}^{(m)}$, $\hat{\mu}_{jk}^{(m)}$, $\hat{\mu}_{l}^{(m)}$, $\hat{\sigma}_{jk}^{2}{}^{(m)}$ and $\hat{\sigma}_{l}^{2(m)}$;

(b) Calculate $\hat{\pi}_{jk}^{(m+1)}$, $\hat{\mu}_{jk}^{(m+1)}$, $\hat{\mu}_{l}^{(m+1)}$, $\hat{\sigma}_{jk}^{2}{}^{(m+1)}$ and $\hat{\sigma}_{l}^{2(m+1)}$ in (4.17)-(4.21);

(c) Calculate $log.value^{(m+1)}$ using $\hat{\pi}_{jk}^{(m+1)}$, $\hat{\mu}_{jk}^{(m+1)}$, $\hat{\mu}_{l}^{(m+1)}$, $\hat{\sigma}_{jk}^{2}{}^{(m+1)}$ and $\hat{\sigma}_{l}^{2(m+1)}$;

(d) If $log.value^{(m+1)}$ is finite and $\frac{log.value^{(m+1)} - log.value^{(m)}}{log.value^{(m)}} < 0.0001$, stop;
If $log.value^{(m+1)}$ is infinite, stop, and return "The Algorithm Converged to A Degenerate Solution";
otherwise go to (a);

Here, `trial` is a user-specified constant, which means how many sets of starting values one wants to use. In the thesis, `trial` $= 100$ is employed. The initial values of $\hat{z}_{ijk}^{(0)}$ are either 0 or 1. Step 3 is the conventional EM algorithm. Thus the multi-step EM can be considered as a sophisticated technique for finding good starting values for the EM algorithm.

Before the beginning of Step 2, a tuning parameter `multiplier` is introduced to adjust small estimates of variances, which usually result in singularity (degenerate solutions). The label assignment of K-means is very sensitive to the outliers, each of which can be a cluster with only a single point. The role of `multiplier` is to enlarge

small variances and recruit more points for small clusters. In the thesis, `multiplier` initially takes values of $2, 3, 4$. For each `multiplier`, the three steps (Step 1 to Step 3) in the Multi-step EM algorithm are run to identify the best parameter estimates. If the algorithm could not converge to a local optima after running all `multiplier`'s provided, then Step 2 is re-run with new `multiplier` value equal to $1.5 \times \texttt{multiplier}^c$, where $\texttt{multiplier}^c$ is the current one (in our experiment, we begin with $\texttt{multiplier}^c = 4$), then so on until the algorithm converges to a local optimum. So a possible sequence of `multiplier` can be $2, 3, 4, 6, 9, 13.5, \ldots$. The `multiplier`s are no larger than 200. The algorithm will return a warning "converged to a degenerate solution" if all `multiplier`'s fail to result in good solutions.

Step 2 is designed to avoid degenerate solutions. Fixing the enlarged local and global variances reduces the possibility that the algorithm moves towards a singular solution too early and allows the algorithm to explore more parameter space before converging to an estimate.

Why do we use K-means as a strategy to choose the starting values for the multi-step EM algorithm? Initially, two kinds of strategies, hierarchical clustering (Fraley & Raftery 2002) and K-means, were compared on the basis of their ability to select better starting values. Using the starting values chosen by K-means, the Multi-step EM algorithm always gave better performance on testing sets than using starting values chosen by hierarchical clustering. In this experiment, the data were simulated from the CMDA1 model, and "performance" was measured using Average Hit Rate, a ranking measurement that will be discussed later in Section

4.8.1. Therefore, K-means is used as the strategy to choose starting values in the thesis.

When applying the Multi-step EM algorithm to real data sets in drug discovery, outliers can still cause difficulties with degenerate solutions. We adapt the Multi-step EM algorithm so that in each iteration outliers are identified and removed. For example, if an observation contributes large log likelihood (usually ∞), this observation can be viewed as an outlier and removed from the data set. Here the outliers are not due to any type of measurement errors, and they are just compounds with very different structure from other compounds. Here, an outlier is defined operationally, i.e. the algorithm behaviour determines what an outlier is. They are likely to be far from other points, but they are defined in terms of likelihood contributions and degeneracy. In the remaining of the thesis, all outliers are defined the same way.

4.6.1 Illustrative Example

A two-dimensional data set simulated from the CMDA1 model is presented to help us understand how the Multi-step EM algorithm proceeds from one step to the next one and the influence of the initial values chosen by K-means. The data are plotted in Figure 4.3. In this example, only one `multiplier` is used, i.e. `multiplier` $= 2$.

Choosing the initial values of $\hat{z}_{ijk}^{(0)}$, i.e. the probability that observation i belongs to the j^{th} component of the k^{th} class, is handled in Step 1 of the Multi-step EM algorithm (Section 4.6). In Chapter 5, we prove that the CMDA1 model is identifiable and does not have the label switching problem. The cluster labels of

observations from K-means are considered as the initial values of z_{ijk}. Since label-switching can occur with K-means, the cluster labels may not correspond to the component labels of the CMDA1 model. In a regular mixture model, swapping labels will not change the model, but it is not true for the CMDA1 model as cluster j has cluster-specific parameters for x_j.

We illustrate convergence of the multi-step EM algorithm in two scenarios, corresponding to whether there is a good match between the initial values from K-means and the final estimates. Figure 4.4 illustrates both good and bad matches between the initial values from K-means. In the right side of Figure 4.4, it is clear that the cluster labels in the real class are switched since the red "•" and "o" plotting symbols are reversed from Figure 4.3.

A Run With "Good" Starting Values from K-means

The convergence of the parameters from the good starting values are plotted in Figure 4.5 (a). The green symbols represent the estimates of μ's (the class specific and global means) and black symbols are the estimates of σ's. The green symbols at the right of the plot are the true means. The lines represent the true variances. In Step 2, the Multi-step EM algorithm searches for optimal location parameters, which do not change very much in Step 3. Then both the location and scale parameters start converging to the true parameters in Step 3. The changes of the log likelihood are plotted in Figure 4.5 (b). The comparisons between the true cluster means and the estimates at various stages of the multi-step EM algorithm are given in Table 4.1. In this case, both the K-means initial values of the μ's and

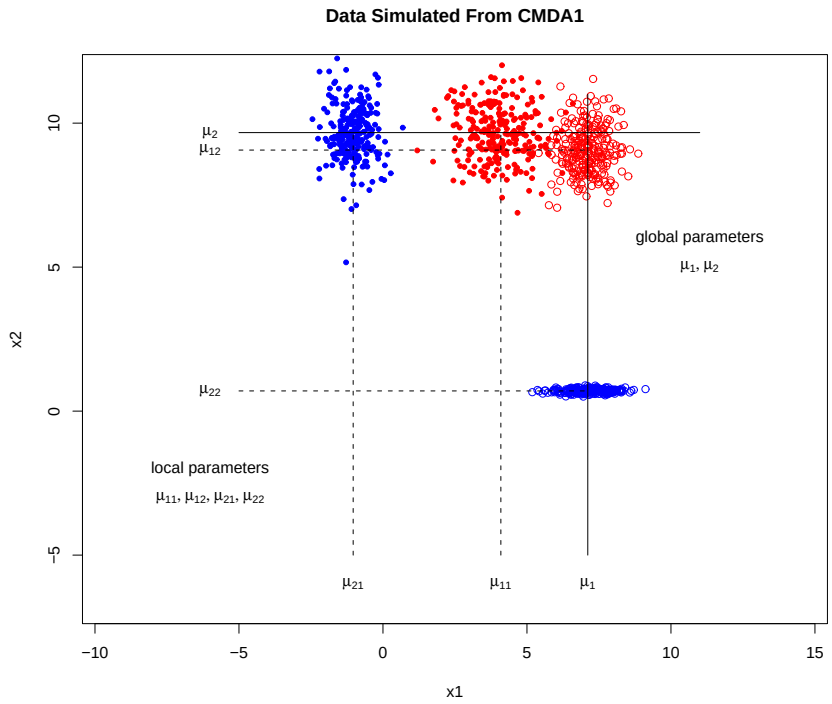

Figure 4.3: Illustrative example of the CMDA1 model in two dimensions. Class is indicated by red/blue, and matching mixture components have the same plotting symbol.

the 1-step EM values are close to the true values.

With "Bad" Starting Values from K-means

When the starting values are not that promising for the Multi-step EM algorithm, it takes the algorithm a little bit longer to converge. The convergence performance of the parameters from the "bad" starting values chosen by K-means is presented in Figure 4.6 (a) and the log likelihood versus the iterations is plotted in Figure 4.6 (b). This example also shows that the Multi-step EM algorithm can automatically

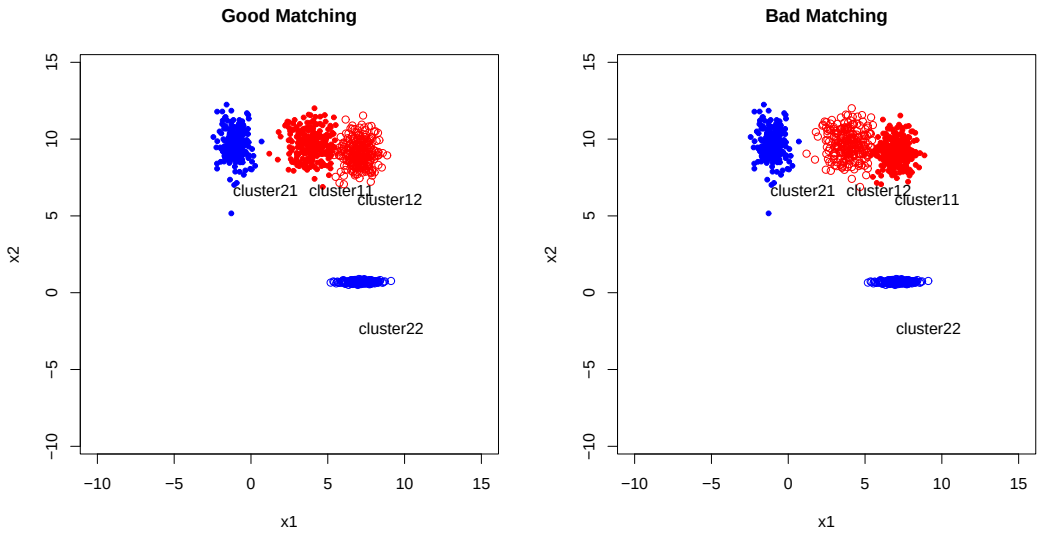

Figure 4.4: Matching of the initial values from K-means: left, a good match; right, a bad match. Plotting symbols and colour are the same as in Figure 4.3.

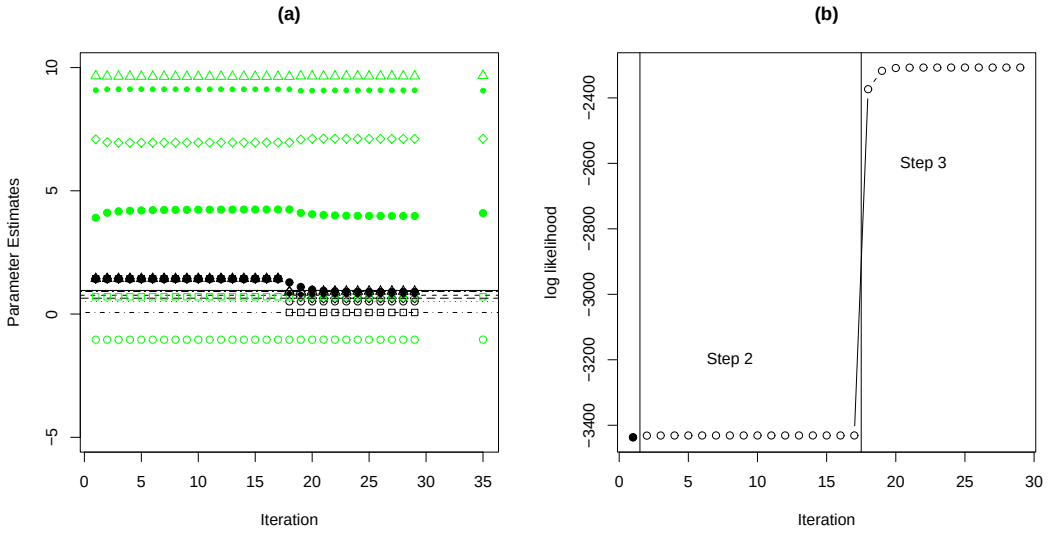

Figure 4.5: when the starting values are "good". (a): Convergence of the parameter estimates; (b): Log likelihood versus iterations. Green symbols on the right of plot (a) represent the means.

		Good Match		Bad Match	
	True cluster mean	Initial K-means centre	Centre for 1EM step	Initial K-means centre	Centre for 1EM step
Cluster 11 (Red •)	(4.09, 9.67)	(3.91, 9.66)	(4.11, 9.64)	(7.04, 9.35)	(6.58,9.41)
Cluster 12 (Red ○)	(7.11, 9.06)	(7.09, 9.08)	(6.97, 9.12)	(5.57, 9.68)	(5.89,9.53)
Cluster 21 (Blue •)	(-1.03, 9.67)	(-1.04, 9.66)	(-1.04, 9.64)	(-1.04, 9.35)	(-1.04,9.41)
Cluster 22 (Blue ○)	(7.11, 0.70)	(7.09, 0.70)	(6.97, 0.70)	(5.57, 0.70)	(5.89,0.70)

Table 4.1: The comparisions between the true cluster means and the centre estimates for both good and bad matches.

adjust the "bad" starting values and converge to the optimal solutions. The "cross-over" of the largest four means in Figure 4.6 (a) is an indication of a correction of initially poor μ values. The comparisons between the true cluster means and the estimates are given in Table 4.1. The $\mathbf{x}$, coordinates of the initial K-means centres are reversed between clusters 11 and 12. After 1 EM step, the estimates are already beginning to move toward the correct values.

This example also illustrates an additional advantage of Step 2. By fixing scale parameters, Step 2 focuses on adjusting location parameters in instances where the initial match is poor.

4.6.2 Parallel Computation

Large drug data sets usually make computations very intensive. In the implementation of the two EM algorithms, parallel computing is employed to speed up computations. A parallel computer is a kind of computer with multiple processors acting to achieve some common goal. In this section, we discuss parallel computing for the EM algorithm.

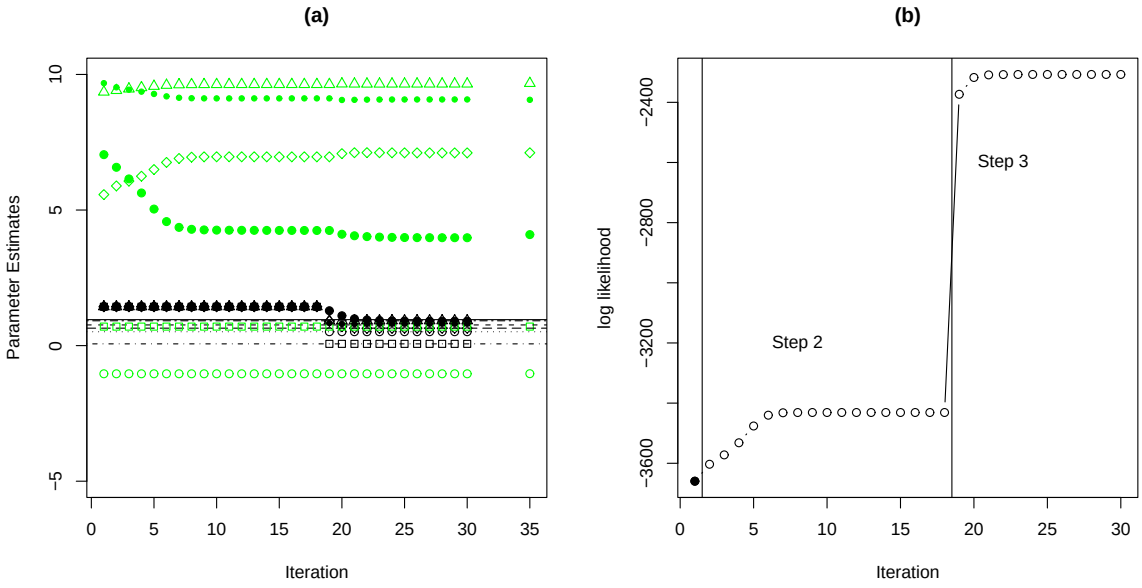

Figure 4.6: when the starting values are "bad". (a): Convergence of the parameter estimates; (b): Log likelihood versus iterations. Green symbols on the right of plot (a) represent the means.

The first question for parallel computing is what can or can not be made parallel? A fundamental issue is whether the code is dependent or not. For example the assignments, $a \Leftarrow b$ and $a \Leftarrow c$ are not parallelizable since the value of a depends on both b and c. However, $a \Leftarrow b$ and $c \Leftarrow d$ are parallelizable as two codes are independent.

Figure 4.7 shows two kinds of parallel computing: (a) embarrassingly parallel and (b) non-embarrassingly parallel. In "embarrassingly parallel" computing, each iteration of code inside a loop is independent of other iterations. Many statistical computations belong to this category, e.g. bootstrapping and cross-validation. In Figure 4.7, the rectangular boxes represent a controlling processor (also called "master") and the circles represent slave processors or slaves. The controlling processor sends jobs to each slave processor and collects results from each slave processor.

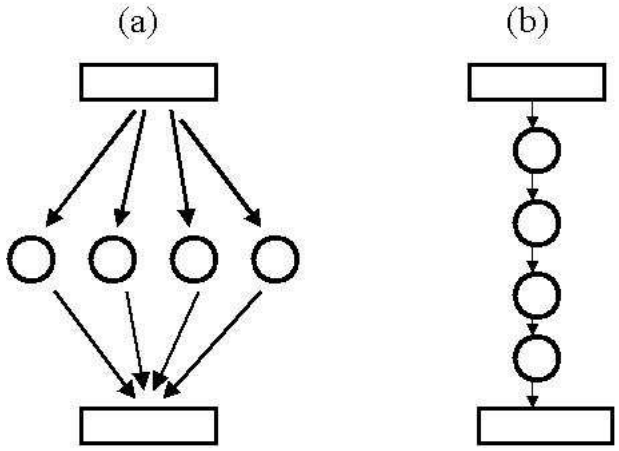

Figure 4.7: Parallel Computing: (a) embarrassingly parallel; (b) non-embarrassingly parallel.

The EM algorithm is not an embarrassingly parallel algorithm since each itera-

tion depends on the previous one as in Figure 4.7 (b). After timing each step in the Multi-step EM algorithm, we find that E-step in the EM algorithm is computationally intensive since this step involves calculation of the membership z_{ijk} for each observation given class k. To determine membership, the normal density function must be evaluated. This is an expensive operation due to calculation of a natural exponent. Therefore, we decide to use the following parallel computing diagram to speed up the process. Figure 4.8 illustrates how the parallel computing works in the EM algorithm. The large data set has been split across each processor, e.g. the first slave processor has observations from 1 to 100, the second slave processor has observations from 101 to 200, etc. At the iteration m, the controlling processor sends the current parameter estimates (μ_{jk}'s, μ_j's, σ_{jk}'s, σ_j's and π_{jk}'s) to each slave in order to calculate cluster labels for observations. Then the slave processors send the values of $\hat{z}_{ijk}$ back to the controlling processor, which will conduct the M-step of the EM algorithm, i.e. calculate the parameter estimates. The process continues until some stopping rule is reached.

Therefore, for jobs that are not embarrassingly parallel, it is possible to do parallelization by having processors communicate data. The speed-up from this parallelization is less than linear, i.e., doubling the number of processors does not make the algorithm run twice as fast. Usually speed-up is either the logarithm of the number of processors or converges towards a value.

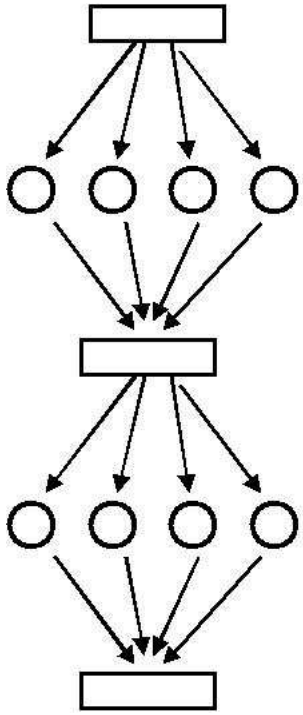

Figure 4.8: Parallel computing for the EM algorithm. Usually, there are more than four processors used in computations, and more than two sending-receiving procedures.

	Factor	Level 1	Level 2
1	Dimensionality	2	10
2	Covariance Structure	same	different
3	Sample Size	small	large
4	Proportion	balanced	unbalanced
5	Mean	well separated	not well separated

Table 4.2: 5 factors and their levels for the CMDA1 model. See text for precise specifications of the levels used.

4.7 Performance of the Multi-step EM Algorithm

In this section, we compare the Multi-step EM algorithm to the EM algorithm when both algorithms are used to estimate the parameters of the CMDA1 model. Here and in the remainder of the thesis, we refer to these algorithms as "Multi-step EM" and "EM". All the simulated data sets used in the remainder of this chapter are assumed to have two classes: active and inactive.

4.7.1 Design of the Simulation

All data will be simulated from CMDA1 models with a variety of parameter settings. Five factors representing important properties of the CMDA1 model are carefully chosen. Each of these factors has two levels. The factors and levels are summarized in Table 4.2.

The interpretations of the five factors, their levels and how to simulate parameters are as follows:

- Dimensionality: the number of descriptors in the data set. In the simulation, either 2 or 10 dimensions are used. When dimensionality= 2, the model is a simple model with only 14 parameters to be estimated. When dimensionality= 10, there are 78 parameters to be estimated. Here, we want to explore the performance of Multi-step EM in two extreme situations.

- Covariance Structure: the within class covariance structures. The CMDA1 model is a mixture of multivariate normal densities with diagonal covariance matrices. Each entry in the covariance matrices is a variance of a univariate normal density function. The within class covariance structures for different components can be the same or different. If Covariance Structure is the same within classes, all the clusters share the same covariance, i.e. $\sigma_1 = \sigma_{11} = \sigma_{21}$, $\sigma_2 = \sigma_{12} = \sigma_{22}$, and $\sigma_3 = \sigma_{13} = \sigma_{23}$ etc. This is the traditional constrained mixture model having a common diagonal covariance matrix. In the simulation, we draw global parameters $\sigma_j \sim U(0.01, 1.5)$, $j = 1, \ldots, P$. When Covariance Structure is different both within and between classes, we draw $\sigma_j \sim U(0.01, 1.5)$ and class specific variances $\sigma_{jk} \sim U(0.01, 1.5)$ for $j = 1, \ldots, P$ and $k = 1, \ldots, K$. Simulated data plotted in Figure 4.9 illustrates the two different levels of Covariance Structure while other factors are the same.

- Sample Size: small ($5 \times$ # of parameters) and large ($10 \times$ # of parameters).

- Proportion: the proportions of active and inactive compounds in the data. When Proportion is balanced, the active and inactive classes have the same number of compounds. If Proportion is unbalanced, the total sample size is

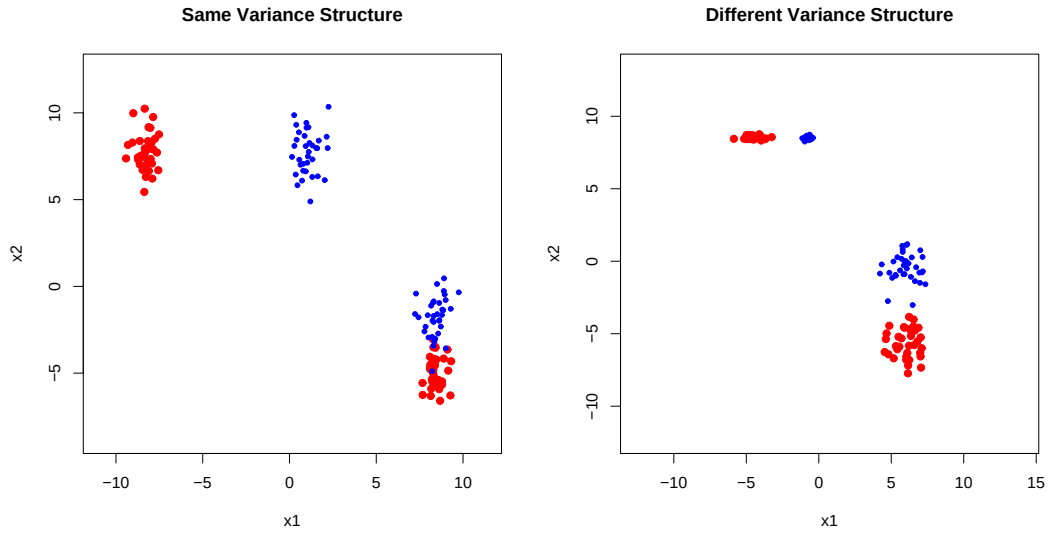

Figure 4.9: Covariance Structure: same (left) and different (right) while other factors are fixed.

divided in a 1 : 9 ratio of active:inactive. The actual numbers of active and inactive compounds in different cases are summarized in Table 4.3.

- Mean: the location parameters for each component. By carefully selecting values for both local and global means, the clusters between classes can be "Well Separated" or "Not Well Separated". When Mean is well separated, we draw $\mu_{j1} \sim U(-9, -3)$, $\mu_{j2} \sim U(-2, 4)$ and $\mu_j \sim U(4, 10)$, $j = 1, \ldots, P$. When Mean is not well separated, we draw $\mu_{jk} \sim U(-3, 3)$ and $\mu_j \sim U(-3, 3)$, $j = 1, \ldots, P$ and $k = 1, \ldots, K$. Figure 4.10 shows two data sets: the left one is well separated and the right one is not well separated.

We note that two of these factors (Covariance and Mean) involve simulations of random values of the parameters of the CMDA1 model. The within-class mixture weights (π_{jk}'s) are set to be equal in all cases.

			Sample Size		
Dimension	Size	Proportion	Active	Inactive	Total
2	Small	1 : 1	35	35	70
2	Small	1 : 9	7	63	70
2	Large	1 : 1	70	70	140
2	Large	1 : 9	14	126	140
10	Small	1 : 1	195	195	390
10	Small	1 : 9	39	351	390
10	Large	1 : 1	390	390	780
10	Large	1 : 9	78	702	780

Table 4.3: The number of active and inactive compounds generated in the simulation for all combinations of Dimension, Sample Size and Proportion.

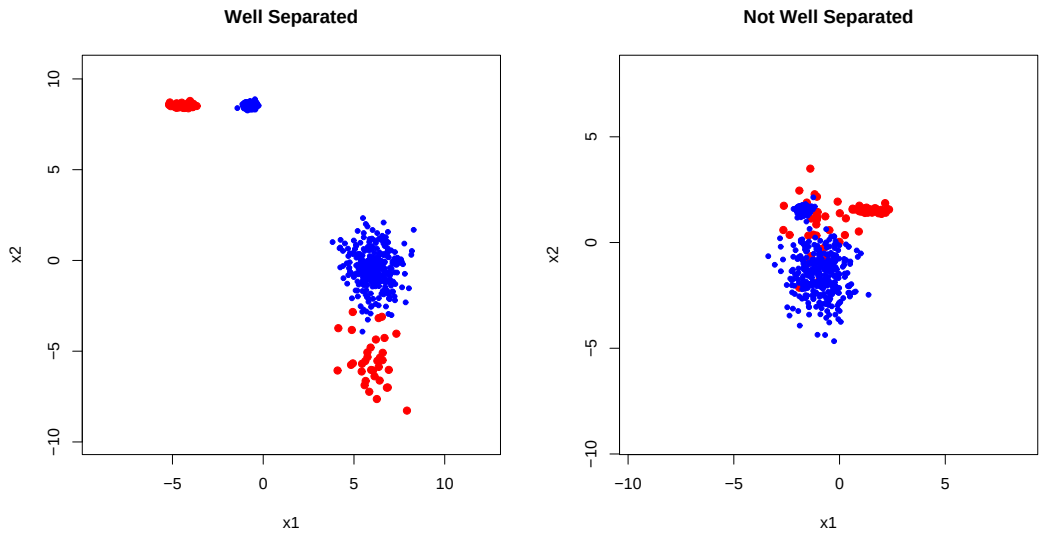

Figure 4.10: Clusters between two classes are Well Separated (left) or Not Well Separated (right) while other factors are fixed.

A full-factorial experiment in these five factors, each of which has two levels, gives us a total of 32 combinations listed in Table 4.4.

In order to effectively compare Multi-step EM to EM, the total number of starting sets for EM is taken as the product of the number of initial value sets (`trials`) and the number of `multiplier`'s used in Multi-step EM. Therefore, the total number of starting sets for EM varies, as the number of `multiplier`'s used in the Multi-step EM may change in different simulations. By allowing EM and Multi-step EM a comparable number of restarts, we hope that differences in performance are mostly due to fixing σ's and using `multiplier`'s in Step 2.

The experiment is carried out as follows:

- For each combination from 1 to 32 in Table 4.4, a model is simulated. These models may share some parameters. For instance, combination 1 and 3 have exactly the same numeric values of μ's and σ's.
- Holding the models fixed, 200 data sets are generated from each model according to combinations 1 to 16. For combinations 17 to 32, 20 data sets are simulated from each model to reduce the amount of computation. The number of data sets is chosen to give small variation in average results, while being computationally feasible.

We have in total 32 runs, and for each run, there are multiple replicates. We emphasize that a model is randomly sampled from each combination first, and then the data sets are independently generated from each model. There is not considerable variability in sampling a model due to our specified simulation procedure.

	Dimension	Covariance	Size	Proportion	Means
1	2	Same	Small	Balanced	Well Separated
2	2	Same	Small	Balanced	Not Well Separated
3	2	Same	Small	Unbalanced	Well Separated
4	2	Same	Small	Unbalanced	Not Well Separated
5	2	Same	Large	Balanced	Well Separated
6	2	Same	Large	Balanced	Not Well Separated
7	2	Same	Large	Unbalanced	Well Separated
8	2	Same	Large	Unbalanced	Not Well Separated
9	2	Different	Small	Balanced	Well Separated
10	2	Different	Small	Balanced	Not Well Separated
11	2	Different	Small	Unbalanced	Well Separated
12	2	Different	Small	Unbalanced	Not Well Separated
13	2	Different	Large	Balanced	Well Separated
14	2	Different	Large	Balanced	Not Well Separated
15	2	Different	Large	Unbalanced	Well Separated
16	2	Different	Large	Unbalanced	Not Well Separated
17	10	Same	Small	Balanced	Well Separated
18	10	Same	Small	Balanced	Not Well Separated
19	10	Same	Small	Unbalanced	Well Separated
20	10	Same	Small	Unbalanced	Not Well Separated
21	10	Same	Large	Balanced	Well Separated
22	10	Same	Large	Balanced	Not Well Separated
23	10	Same	Large	Unbalanced	Well Separated
24	10	Same	Large	Unbalanced	Not Well Separated
25	10	Different	Small	Balanced	Well Separated
26	10	Different	Small	Balanced	Not Well Separated
27	10	Different	Small	Unbalanced	Well Separated
28	10	Different	Small	Unbalanced	Not Well Separated
29	10	Different	Large	Balanced	Well Separated
30	10	Different	Large	Balanced	Not Well Separated
31	10	Different	Large	Unbalanced	Well Separated
32	10	Different	Large	Unbalanced	Not Well Separated

Table 4.4: The 32 combinations of the five factors with two levels.

Run	1	2	3	4	5	6	7	8
Multi-step EM	0	0	0	20	0	0	0	1
EM	103	200	174	200	3	200	44	199
Run	9	10	11	12	13	14	15	16
Multi-step EM	0	1	0	14	0	0	0	1
EM	24	145	77	157	0	68	1	83

Table 4.5: Degenerate solutions from the Multi-step EM and EM algorithms. In runs 17-32, all EM solutions are degenerate.

In the following sections, three comparisons between Multi-step EM and EM will be made: degeneracy, parameter estimation and prediction accuracy via the likelihood.

4.7.2 Degenerate Solutions

Compared to EM, one advantage of Multi-step EM is significantly reducing the possibility that the algorithm converges to degenerate solutions. When the dimensionality is 2, the hardest classification problem is probably the 12^{th} combination, i.e. the covariance structure is "Different", the sample size is "Small", the data are "Unbalanced" and the clusters from different classes are "Not Well Separated". For combination 12, Multi-step EM converges 14 times to degenerate solutions, while EM converges 157 times over 200 replicates. The number of degenerate solutions for combinations 1-16 in Table 4.4 are summarized in Table 4.5.

From Table 4.5, although Multi-step EM sometimes still converges to degenerate solutions, it has significantly reduced the amount of degenerate solutions. The

Run	17	18	19	20	21	22	23	24
Multi-step EM	0	2	0	13	0	0	0	1
Run	25	26	27	28	29	30	31	32
Multi-step EM	0	1	0	16	0	1	0	1

Table 4.6: Degenerate solutions from the Multi-step EM for runs 17-32, while all EM solutions are degenerate.

same experiments are also applied to high-dimensional cases (combinations 17 to 32). However, EM fails completely, always converging to degenerate solutions. The degenerate solutions from the Multi-step EM algorithm for runs 17-32 are summarized in Table 4.6. Thus, runs 17-32 will not be studied further in subsequent comparisons.

4.7.3 Parameter Estimation Accuracy

We first consider the 200 replicates of model 12. Figure 4.11 displays histograms of the four local standard deviation estimates $\hat{\sigma}_{jk}$, $j = 1, 2$, and $k = 1, 2$, obtained via Multi-step EM and EM. Degenerate solutions are included in this plot, and correspond to one or more $\hat{\sigma}_{jk}$ being equal to zero. Figure 4.12 is similar but for global $\hat{\sigma}_j$'s, $j = 1, 2$. Figure 4.11 and Figure 4.12 indicate that Multi-step EM gives more accurate parameter estimates than EM, since the estimates are generally closer to the true values. The Multi-step EM estimates for both local and global standard deviations have less variance than those estimated by EM. Moreover, for most of the standard deviations, the estimates from EM are biased.

Mean square errors (MSE) for all the parameter estimates in combination 12

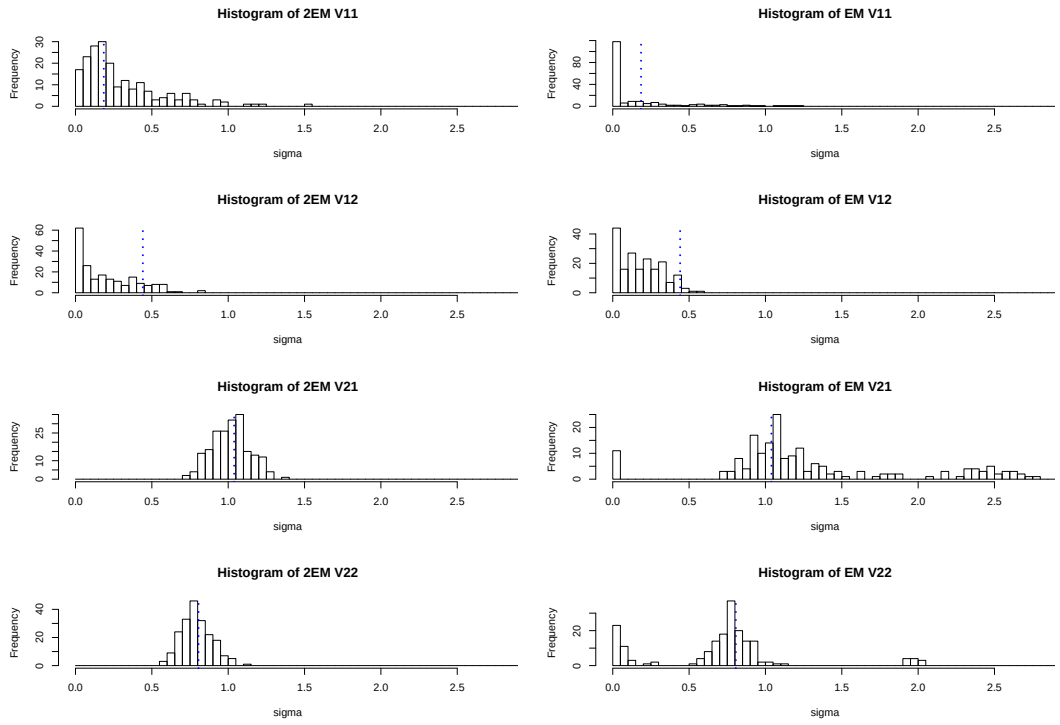

Figure 4.11: The estimates of four local standard deviations ($\hat{\sigma}_{jk}$) estimated by Multi-step EM (left column) and EM (right column) based on 200 realizations of model 12. The vertical dotted lines indicate the true parameter values.

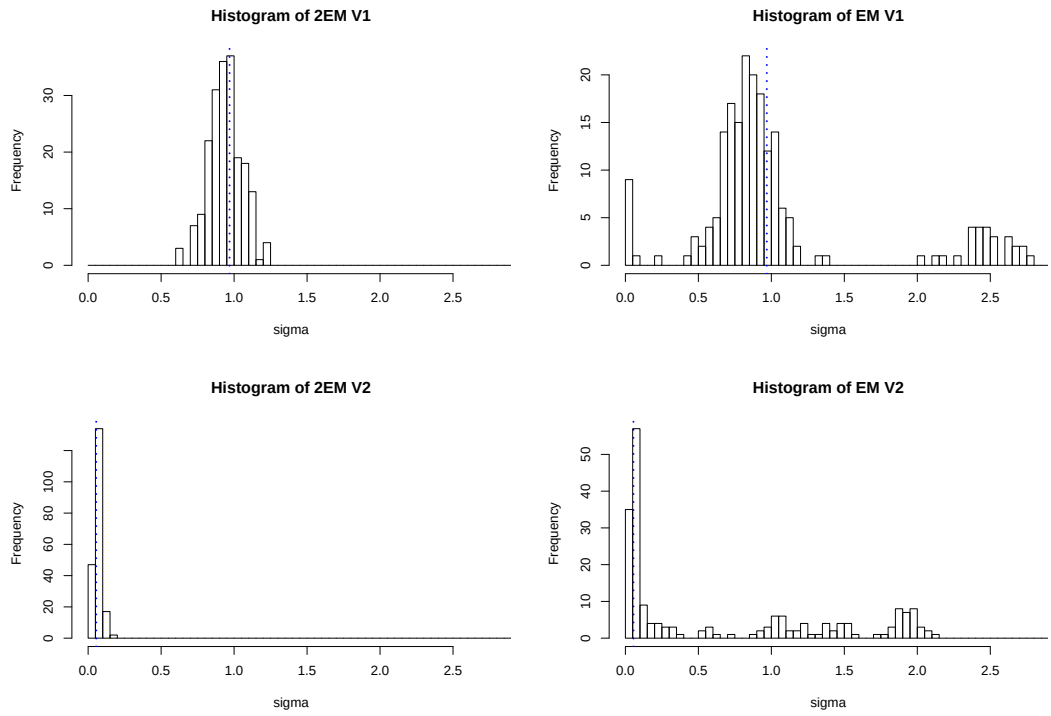

Figure 4.12: The estimates of two global standard deviations ($\hat{\sigma}_j$) estimated by Multi-step EM (left column) and EM (right column) based on 200 realizations of model 12. The vertical dotted lines indicate the true parameter values.

Parameters	Multi-step EM	EM
μ_{11}	0.09181	1.64753
μ_{12}	0.11417	0.07589
μ_{21}	0.03222	4.98514
μ_{22}	0.01971	1.73415
μ_1	0.03014	2.35141
μ_2	0.00014	1.55863
σ_{11}	0.08255	0.07405
σ_{12}	0.09484	0.08572
σ_{21}	0.01559	0.50813
σ_{22}	0.01026	0.29383
σ_1	0.01366	0.39069
σ_2	0.00065	0.91055

Table 4.7: The estimated MSEs for the parameters of combination 12, over 200 realizations. Degenerate solutions are included in these calculations.

are listed in Table 4.7.

The MSE's for the estimates of σ_{11} of the first 16 low-dimensional cases are plotted in Figure 4.13, which shows that Multi-step EM gives more accurate parameter estimates than EM overall. Plots for other 13 parameters (not shown here) are similar.

4.7.4 Prediction Accuracy via Log Likelihood

We also compare the algorithms in terms of their prediction ability. Usually we would like to know the performance of our model on some testing sets, as it is a good measurement of how well a model predicts future data sets. Using the

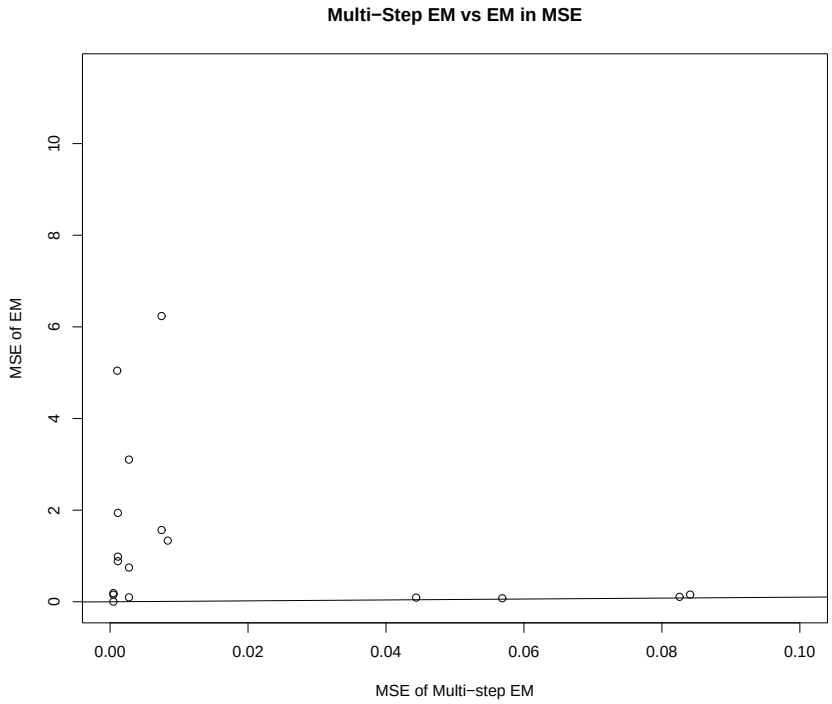

Figure 4.13: The plot of MSE's for the estimates of σ_{11} for combinations 1 to 16. The x-axis and y-axis are in different scales. The line in the plot is the 45 degree line, on which the MSE's are equal.

test set, we will evaluate $l_n(\hat{\mathbf{\Psi}})$, the log likelihood of the parameter values, estimated from the training set via EM or Multi-step EM. An additional advantage of using the test set is that unless one observes a test point exactly equal to a training point, $l_n(\hat{\mathbf{\Psi}})$ will be less than ∞, enabling us to access the quality of prediction for both degenerate and non-degenerate solutions. A large testing set ($100 *$ sample size of the training data) is generated.

For combination 12 in Table 4.4, the differences of the log likelihood from the log likelihood of the true parameters for the 200 testing sets are calculated using the estimates from EM and Multi-step EM respectively. The testing size in this case is $7,000$. The differences are obtained by subtracting the true log likelihood from the log likelihood estimated by each algorithm. The difference of the log likelihood is denoted by

$$\Delta = l_n(\hat{\mathbf{\Psi}}) - l_n(\mathbf{\Psi_0}). \tag{4.24}$$

We expect $\Delta < 0$ since $\hat{\mathbf{\Psi}}$ maximizes the training likelihood, not the test likelihood. The smaller the absolute difference is, the closer the estimated log likelihood is to the truth. Figure 4.14 shows Δ pairs corresponding to Multi-step EM and EM. The absolute differences of log likelihood calculated by the Multi-step EM algorithm are usually smaller than those calculated by the EM algorithm. In the scatter plot of Figure 4.14, the line represents equal performance; points below the line mean Multi-step EM is better than EM; those above the line mean Multi-step EM is worse than EM. There are 150 points out of 200 under the line, i.e., Multi-step EM has better prediction ability than EM. The frequency of such "wins" of the Multi-step EM for combinations 1-16 are summarized in Table 4.8. For all 16

low-dimensional combinations in Table 4.4, the same kind of pattern can be found in the differences of the log likelihood compared between EM and Multi-step EM.

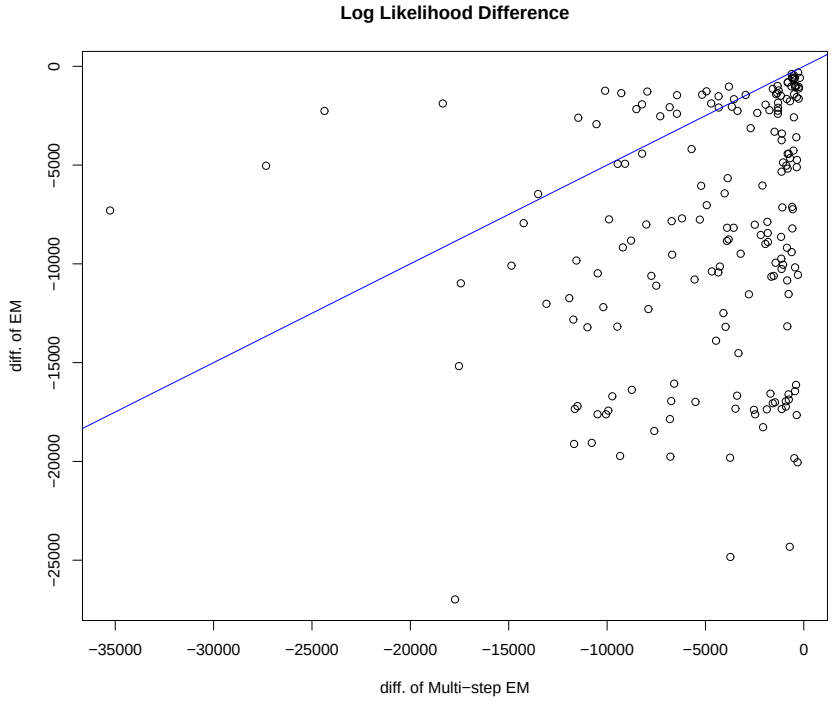

Figure 4.14: A comparison log likelihood differences for EM and Multi-step EM based on 200 testing sets generated from the model 12. The line is a 45 degree line. The plot includes degenerate solutions.

4.7.5 Conclusions for the Multi-step EM Algorithm

From the above experiments, Multi-step EM has the following advantages compared to EM:

- Multi-step EM can give accurate estimates for the parameters of the CMDA1 model.

Combination	1	2	3	4	5	6	7	8
Wins	135	169	174	135	102	196	133	197
Combination	9	10	11	12	13	14	15	16
Wins	156	156	178	150	177	176	193	154

Table 4.8: Number of replicates (out of 200) in which Multi-step EM has test set log likelihood that is superior to EM, for combinations 1-16.

- The estimates from Multi-step EM have lower MSE's than those estimated by EM.
- Multi-step EM is superior to EM in the prediction ability of the estimated model.
- Multi-step EM significantly reduces the number of degenerate solutions.

Since Multi-step EM still cannot totally avoid degeneracy, another approach, Penalized Maximum Likelihood Estimation (PMLE), is considered in the next chapter.

4.8 Performance as Classifiers: CMDA1 vs. MclustDA

In this section, the CMDA1 model will be compared to the popular model-based discriminant approach, MclustDA (Fraley & Raftery 2002). In MclustDA, the density of $\mathbf{x}$ within each class is modeled as a mixture of multivariate normal densities. In the form of MclustDA used here, we assume the multivariate normal mixture components have diagonal covariance matrices, and allow both the mean

vector and the covariance matrix to vary across mixture components. CMDA1 will be estimated via the multi-step EM algorithm described in Section 4.6. Methods will be compared on the basis of their predictive performance as classifiers.

Two scenarios are carefully considered to compare the CMDA1 model and the MclustDA model: (1) the true model is the CMDA1 model (Section 4.8.2), and (2) the true model is the MclustDA model (Section 4.8.3). For both scenarios, the comparison between the CMDA1 model and the MclustDA model focuses on prediction accuracy for testing sets. In order to compare the CMDA1 model and the MclustDA model on a fair basis, the number of components in each model is made the same as P (the dimensionality of the data). For example, if the dimensionality of the data is 6, the number of components in both of the CMDA1 and the MclustDA models is 6.

The next section introduces the Average Hit Rate (AHR), a performance criterion for unbalanced data.

4.8.1 Comparison Criteria: Misclassification Rate and Average Hit Rate

Consider the common classification problem with 2 classes $y \in \{0, 1\}$ (or the active and inactive classes in drug discovery). Often, the misclassification rate and log likelihood (or the equivalent measurement, deviance) are used as criteria for model building (with training data) or for model assessment (with testing data). The misclassification rate is simply the proportion of observations assigned to the wrong classes. Since the misclassification rate is more straightforward than the log

likelihood, the misclassification rate is preferred here. Pursuing a low misclassification rate is a common strategy for solving the classification problem. However, the misclassification rate is not always an appropriate standard, especially in drug discovery, where the active compounds are rare. When the proportion of active compounds in the test data is small, even a "bad" classifier, which classifies all active compounds in the testing set as inactive, still can return a small misclassification rate. The reason is that the misclassification method is easily dominated by the majority groups of data set. Hence the misclassification rate is not a reasonable model assessing criterion for the rare target problems. We will use misclassification rate, but only in the scenarios where response classes are balanced.

Wang (2005) developed the Average Hit Rate (AHR) as a criterion for drug discovery problems. AHR measures the ability of classifiers to give the best ranking of compounds. Suppose n compounds have been ranked according to some measure of how likely a compound is to be active. Denote the response values for such an ordered list of compounds by $y_{(1)}, \ldots, y_{(n)}$, with $y_{(1)}$ corresponding to the compound predicted to be the most likely to be active. $y_{(i)}$ equals 1 if the i^{th} compound is active, otherwise 0. The AHR is defined as

$$AHR = \frac{\sum_{i=1}^{n} \frac{y_{(i)} \sum_{l=1}^{i} y_{(l)}}{i}}{A} \tag{4.25}$$

where $A = \sum_{i=1}^{n} y_{(i)}$, i.e. the number of actives in the list. Table 4.9 is a simple example to show how to calculate AHR. There are three different rankings. The best ranking ranks all three active compounds before the two inactive ones, so the AHR of this ranking is 1 (i.e. $\frac{1\times\frac{1}{1}+1\times\frac{2}{2}+1\times\frac{3}{3}+0\times\frac{3}{4}+0\times\frac{4}{5}}{3} = 1$). The worst ranking gives a higher ranking to the inactive compounds than the active ones, yielding

List	Best Ranking $y_{(i)}$	Best Ranking $\sum_{l=1}^{i} \frac{y_{(l)}}{i}$	Middle Ranking $y_{(i)}$	Middle Ranking $\sum_{l=1}^{i} \frac{y_{(l)}}{i}$	Worst Ranking $y_{(i)}$	Worst Ranking $\sum_{l=1}^{i} \frac{y_{(l)}}{i}$
1	1	1	1	1	0	0
2	1	1	1	1	0	0
3	1	1	0	0.67	1	0.33
4	0	0.75	1	0.75	1	0.5
5	0	0.6	0	0.6	1	0.6
AHR	$\frac{1+1+1}{3} = 1$		$\frac{1+1+0.75}{3} = 0.92$		$\frac{0.33+0.5+0.6}{3} = 0.48$	

Table 4.9: Example of calculating average hit rate.

AHR= 0.48. The middle ranking gives one of the active compounds lower priority than one inactive compounds, and then has AHR equal to 0.92. Thus, AHR is a good indicator of how well a classifier ranks active compounds.

In the following sections, misclassification rate is applied only when the simulated data sets are balanced, i.e. the active class has an equal amount of data as the inactive class. The AHR will be used as a comparison criterion when the simulated data sets are imbalanced. In Section 4.9, the AHR will be used in the NCI Antiviral AIDS data, in which active compounds are only 2% of the whole data set.

4.8.2 Data Simulated from CMDA1

Experimental Design

The 32 different CMDA1 models used in Section 4.7 are also used here to generate data. For the combinations with balanced samples (runs 1, 2, 5, 6, 9, 10, 13, 14, 17,

18, 21, 22, 25, 26, 29, 30), the misclassification rate is used to measure performance. For the other 16 unbalanced combinations, AHR is the performance measurement.

Results

Table 4.10 summarizes the average misclassification rates and their standard errors for Bayes', the CMDA1 model and the MclustDA model. In the Bayes' misclassification rate, predictions are generated using the true values of all model parameters. Table 4.11 summarizes the average AHR and their standard errors for Bayes', the CMDA1 model and the MclustDA model. In both tables, the average performance measurement (misclassification rate or AHR) is calculated over 200 samples for two-dimensional data, and 20 samples for 10-dimensional data. In all combinations, CMDA1 is superior to MclustDA, with a lower misclassification rate (Table 4.10) and a higher AHR (Table 4.11).

A two-sample t-test for equal means is conducted to test if there is a significant performance difference between CMDA1 and MclustDA for average misclassification rate. In this test, unequal variances are assumed. For the average misclassification rate (see Table 4.12), there is a statistically significant difference between CMDA1 and MclustDA at a 1% significance level, which indicates CMDA1 performs significantly better than MclustDA in both low and high dimensional balanced cases.

The same t-test with unequal variances assumed is also conducted for the average AHR's (see Table 4.11, the unbalanced model) between CMDA1 and MclustDA. The same conclusion can be made: CMDA1 has a significantly better performance

	Bayes		CMDA1		MclustDA	
Combination	Ave. Mis. Rate	(se)	Ave. Mis. Rate	(se)	Ave. Mis. Rate	(se)
1	0.00	(0.00)	0.00	(0.00)	0.05	(0.00)
2	4.21	(0.02)	4.86	(0.09)	5.43	(0.08)
5	0.00	(0.00)	0.00	(0.00)	0.05	(0.00)
6	4.19	(0.01)	4.41	(0.02)	4.63	(0.01)
9	0.17	(0.00)	0.24	(0.00)	0.38	(0.00)
10	3.79	(0.02)	4.15	(0.00)	5.28	(0.07)
13	0.17	(0.00)	0.20	(0.00)	0.23	(0.01)
14	3.80	(0.01)	3.94	(0.02)	4.29	(0.04)
17	0.97	(0.00)	1.09	(0.00)	4.30	(0.00)
18	12.47	(0.00)	19.46	(0.14)	24.78	(0.15)
21	0.97	(0.01)	1.03	(0.01)	1.19	(0.02)
22	12.52	(0.03)	20.77	(0.18)	23.07	(0.09)
25	0.01	(0.00)	0.04	(0.00)	0.52	(0.00)
26	13.23	(0.00)	15.99	(0.06)	21.15	(0.18)
29	0.01	(0.00)	0.02	(0.00)	0.12	(0.00)
30	13.23	(0.02)	20.02	(1.27)	19.96	(0.27)

Table 4.10: Average Misclassification Rate (%) calculated for Bayes, CMDA and MclustDA when the true model is the CMDA1 model. Standard errors are given in parentheses.

	Bayes		CMDA1		MclustDA	
Combination	Ave. AHR	(se)	Ave. AHR	(se)	Ave. AHR	(se)
3	100	(0.00)	99.8	(0.12)	95.4	(0.68)
4	86.6	(0.08)	79.4	(0.54)	64.3	(0.09)
7	100	(0.00)	100	(0.00)	99.5	(0.00)
8	86.6	(0.05)	84.6	(0.16)	81.2	(0.28)
11	99.9	(0.00)	99.4	(0.20)	92.1	(0.85)
12	92.6	(0.04)	87.7	(0.48)	73.7	(1.14)
15	99.9	(0.00)	99.9	(0.00)	99.8	(0.01)
16	92.6	(0.03)	91.4	(0.17)	89.3	(0.21)
19	98.9	(0.01)	96.8	(0.43)	91.1	(0.12)
20	67.2	(0.04)	46.3	(1.32)	26.9	(0.87)
23	98.9	(0.00)	98.6	(0.05)	95.1	(0.11)
24	67.4	(0.02)	47.7	(1.09)	41.0	(0.90)
27	100	(0.00)	99.3	(0.35)	94.43	(0.08)
28	79.7	(0.02)	58.6	(0.32)	52.1	(0.68)
31	99.99	(0.00)	99.99	(0.00)	99.81	(0.04)
32	79.64	(0.01)	64.13	(0.24)	55.01	(0.39)

Table 4.11: Average AHR calculated (%) for Bayes, CMDA and MclustDA when the true model is the CMDA1 model. Standard errors are given in parentheses.

than MclustDA for unbalanced data.

4.8.3 Data Simulated from MclustDA

Experimental Design

When the true model is the MclustDA model, we perform a different experiment. Five factors, each of which has two levels, are chosen to represent the properties of the MclustDA model. The five factors and their levels are listed in Table 4.14. Here, the MclustDA model shares some of the same assumptions as the CMDA1 model:

- The number of clusters in each class is the same as the dimensionality of the data.
- The independence assumption of the descriptors still holds for the MclustDA model, i.e. the within class covariance matrices are diagonal.

Most of the five factors and their levels are the same as in Section 4.7.1, except Covariance Structure is constructed differently for MclustDA. Unlike CMDA, there are no global parameters used in MclustDA. The interpretations for these factors and their levels are:

- `Dimensionality`: the number of descriptors in the data set. There are two choices for Dimensionality: either 2 and 10. As in the CMDA1 model, we assume that the number of components in each class is equal to the dimensionality. When the dimensionality is 2, there are two clusters in each class, while for 10-dimensional data, there are 10 clusters in each class.

	t-test	
Combination	P-value	Significance
1	< 2.2e-16	***
2	2.070e-06	***
5	< 2.2e-16	***
6	8.66e-10	***
9	1.683e-10	***
10	< 2.2e-16	***
13	8.258e-05	***
14	1.321e-15	***
17	< 2.2e-16	***
18	0.006858	**
21	1.048e-06	***
22	2.801e-04	***
25	6.516e-09	***
26	7.461e-08	***
29	< 2.2e-16	***
30	0.001086	**

Table 4.12: Two sample t-test for CMDA and MclustDA when the true model is the CMDA1 model. Differences in mean AHR significant at the 5%, 1%, and 0.1% levels are denoted by *, **, and ***, respectively.

	t-test	
Combination	P-value	Significance
3	1.659e-09	***
4	< 2.2e-16	***
7	< 2.2e-16	***
8	< 2.2e-16	***
11	8.096e-15	***
12	< 2.2e-16	***
15	2.975e-11	***
16	1.265e-13	***
19	4.799e-12	***
20	8.748e-14	***
23	< 2.2e-16	***
24	2.964e-05	***
27	8.619e-12	***
28	0.003107	**
31	8.387e-05	***
32	0.001178	**

Table 4.13: Two sample t-test for CMDA1 and MclustDA when the true model is the CMDA1 model. Differences in mean AHR significant at the 5%, 1%, and 0.1% levels are denoted by *, **, and ***, respectively.

	Factor	Level 1	Level 2
1	Dimensionality	2	10
2	Covariance Structure	same	different
3	Sample Size	small	large
4	Proportion	balanced	unbalanced
5	Mean	well separated	not well separated

Table 4.14: Five factors and their levels for the MclustDA model.

- Covariance Structure: the within and between class covariance structures. Under our independence assumption, the covariance matrices are diagonal. Two extreme situations are considered here: the covariance matrices are either the same or different within and between classes. If Covariance Structures are the same within and between classes (i.e. $\mathbf{\Sigma}_{jk} = \mathbf{\Sigma}$, for $j = 1, \ldots, P$ and $k = 1, \ldots, K$), a vector with a length being equal to the dimensionality is randomly sampled from a uniform distribution $U(0.01, 2)$, and then the values of the vector are assigned to the diagonal values of the covariance matrix $\mathbf{\Sigma}$. When Covariance Structures are different, the diagonal vectors of all covariance matrices will be independently and randomly sampled from the same uniform distribution $U(0.01, 2)$. Figure 4.15 illustrates these different scenarios.

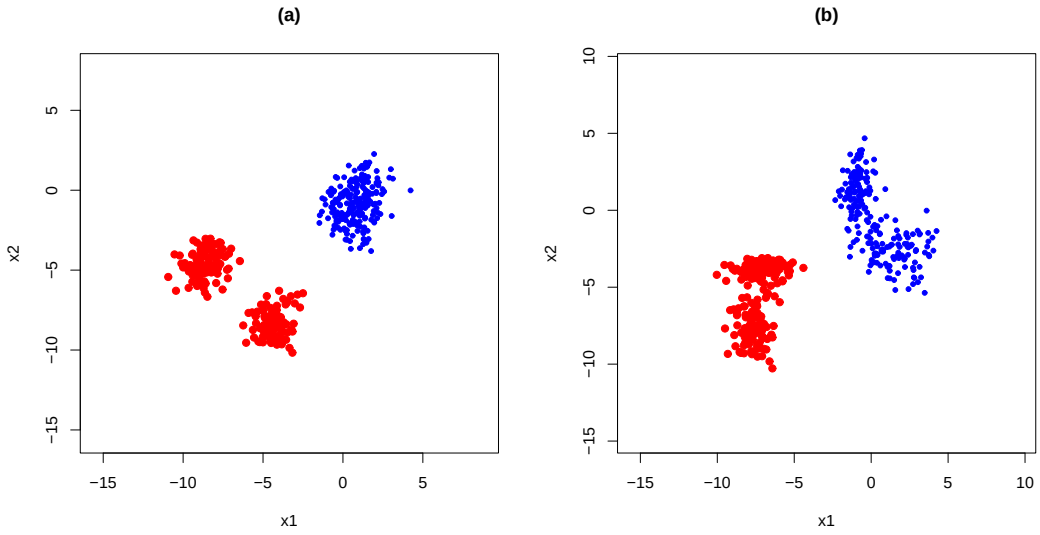

Figure 4.15: Variance Structure: (a) same or (b) different while other factors are the same.

- Sample Size: small ($5 \times$ # of parameters) and large ($10 \times$ # of parameters).

- When Proportion is balanced, the active and inactive classes have the same number of compounds. If Proportion is unbalanced, the active class has only 1/10 of the whole data set. The actual numbers of active and inactive compounds in different cases are summarized in Table 4.15.

- `Mean`: the mean vector for each cluster. By carefully selecting values of mean vectors, the classes can be either "Well Separated" or "Not Well Separated". When Mean is well separated, the elements of mean vectors for the active class will be independently sampled from $U(-9, -3)$ and those for the inactive class will be independently sampled from $U(-3, 3)$. When Mean is not well separated, all the means are randomly and independently sampled from $U(-3, 3)$. Figure 4.16 shows two data sets: (a) is well separated and (b) is not well separated.

Results

Here, only the average performance measurements (misclassification rate or AHR) and their standard errors, and hypothesis tests of equal average performance are listed. Both Table 4.16 and Table 4.17 indicate that in the low-dimensional cases (combinations 1-16), MclustDA outperforms CMDA1. In most of the high-dimensional cases, the CMDA1 model is surprisingly superior to the MclustDA model, returning smaller misclassification rates and higher AHR's. We hypothesize that in these cases, MclustDA performs poorly because of the large number of parameters (rel-

			Sample Size		
Dimension	Size	Proportion	Active	Inactive	Total
2	Small	1 : 1	45	45	90
2	Small	1 : 9	9	81	90
2	Large	1 : 1	90	90	180
2	Large	1 : 9	18	162	180
10	Small	1 : 1	1045	1045	2090
10	Small	1 : 9	209	1881	2090
10	Large	1 : 1	2090	2090	4180
10	Large	1 : 9	418	3762	4180

Table 4.15: The number of active and inactive compounds generated in the simulation for all combinations of Dimension, Sample Size and Proportion.

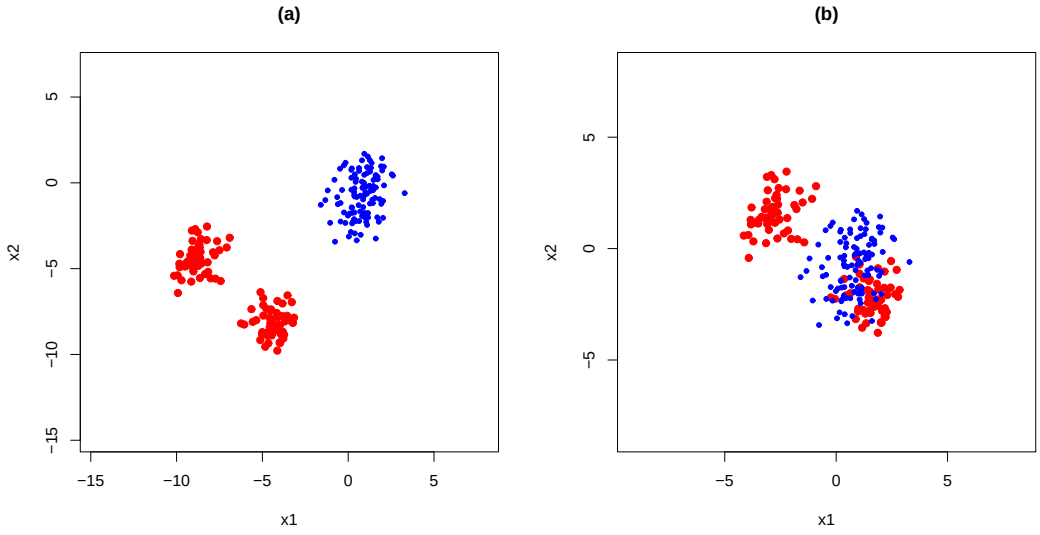

Figure 4.16: Clusters between two classes are: (a) Well Separated or (b) Not Well Separated while other factors are same.

	CMDA1		MclustDA		Difference	t-test	
Comb.	Ave. Mis. Rate	(se)	Ave. Mis. Rate	(se)	CMDA1 vs MclustDA	P-value	Significance
1	0.02	(0.00)	0.00	(0.00)	0.02	0.003555	**
2	18.47	(0.02)	12.88	(0.02)	5.59	< 2.2e-16	***
5	0.00	(0.00)	0.00	(0.00)	0.00	0.003130	**
6	18.32	(0.01)	12.77	(0.01)	5.55	< 2.2e-16	***
9	0.07	(0.00)	0.00	(0.00)	0.07	6.1e-07	***
10	17.99	(0.03)	16.19	(0.03)	1.80	4.683e-15	***
13	0.05	(0.00)	0.00	(0.00)	0.05	3.355e-09	***
14	17.75	(0.02)	15.85	(0.02)	1.90	< 2.2e-16	***
17	0.00	(0.02)	0.00	(0.00)	0.00	0.0002437	***
18	25.23	(0.25)	41.11	(0.01)	-15.88	8.21e-15	***
21	0.11	(0.02)	0.00	(0.00)	0.11	3.054e-05	***
22	25.17	(0.20)	41.04	(0.87)	-15.87	3.397e-14	***
25	0.34	(0.01)	0.44	(0.13)	-0.10	0.4304	
26	25.80	(0.26)	50.29	(0.86)	-24.49	< 2.2e-16	***
29	0.34	(0.00)	0.62	(0.13)	-0.28	0.04395	*
30	25.86	(0.12)	49.76	(0.86)	-23.9	< 2.2e-16	***

Table 4.16: Mean misclassification rate for CMDA1 and MclustDA when the true model is the MclustDA model. Differences in mean misclassification rate significant at the 5%, 1%, and 0.1% levels are denoted by *, **, and ***, respectively.

ative to available data). For example, with 10 dimensions, the CMDA1 model has 78 parameters, while MclustDA model would have 418 parameters.

In such situations, CMDA1 makes more parsimonious use of data and seems to offer superior performance as a result.

	CMDA1		MclustDA		Difference	t-test	
Comb.	Ave. AHR (%)	(se %)	Ave. AHR (%)	(se %)	CMDA1 vs MclustDA	P-value	Significance
3	100	(0.00)	100	(0.00)	0.00	0.99	
4	70.80	(0.19)	80.40	(0.10)	-9.60	5.693e-15	***
7	100	(0.00)	100	(0.00)	0.00	0.99	
8	71.66	(0.09)	81.29	(0.05)	-9.63	< 2.2e-16	***
11	100	(0.00)	99.99	(0.00)	0.01	0.442	
12	53.61	(0.25)	62.57	(0.25)	-8.96	1.469e-09	***
15	100	(0.00)	100	(0.00)	0.00	0.8467	
16	54.08	(0.28)	66.71	(0.10)	-12.63	1.166e-12	***
19	100	(0.00)	100	(0.00)	0.00	0.99	
20	53.82	(1.29)	14.98	(0.49)	38.84	< 2.2e-16	***
23	100	(0.00)	100	(0.00)	0.00	0.03097	*
24	53.82	(1.30)	15.54	(0.84)	38.28	< 2.2e-16	***
27	99.99	(0.00)	99.22	(0.21)	0.77	0.001590	**
28	31.52	(1.08)	10.59	(0.29)	20.93	8.06e-15	***
31	99.99	(0.00)	98.92	(2.67)	1.07	0006772	***
32	31.91	(1.21)	9.58	(1.91)	22.33	7e-14	***

Table 4.17: Average AHR (%) calculated for CMDA and MclustDA when the true model is the MclustDA model. Two sample t-test of mean misclassification rate for CMDA1 and MclustDA when the true model is the MclustDA model. Differences in mean AHR significant at the 5%, 1%, and 0.1% levels are denoted by *, **, and ***, respectively.

Split	CMDA1	MclustDA
1	11.87	2.28
2	13.09	4.41
3	11.34	2.06
4	10.22	7.92
Average	11.63	4.17

Table 4.18: NCI data: AHR (%) calculated for CMDA and MclustDA.

4.9 Application to the NCI Antiviral AIDS Data

We apply both the CMDA1 model and the MclustDA model to the NCI Antiviral AIDS Data. The data set is randomly split using stratified sampling into a training set and a test set, each with $n = 14,906$ compounds, of which 304 are active compounds. We conduct four experiments (4 splits), which will be referred to as "Split 1", ..., "Split 4" in the text below. Performance is assessed by the AHR on the test set. The AHR's returned from both approaches for the 4 splits are listed in Table 4.18. It is clear that the CMDA1 model returns higher AHR's than the MclustDA model. Over the four replications, a paired t-test concludes that the CMDA1 model significantly outperforms MclustDA at a 5% significance level.

We also plot the densities of each BCUT descriptor for the active class and the inactive class in Figure 4.17. It clearly indicates that the estimated means match the centres of the densities.

The mixing proportions estimated from Split 2 are: the active class (π_{11}, π_{21}, π_{31}, π_{41}, π_{51}, π_{61}) $= (0.16, 0.37, 0.05, 0.17, 0.14, 0.12)$ and the inactive class (π_{12}, π_{22}, π_{32}, π_{42}, π_{52}, π_{62}) $= (0.03, 0.09, 0.23, 0.21, 0.23, 0.21)$. The values of these

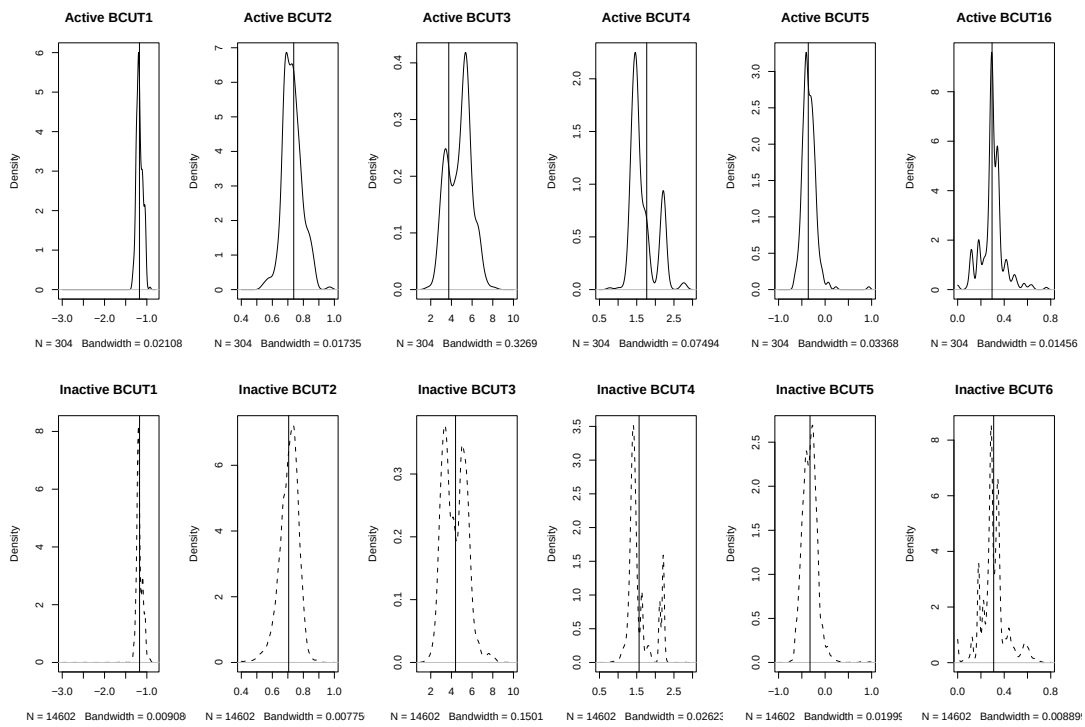

Figure 4.17: The plot of densities of the BCUT descriptors for the active class (the first row) and the inactive class (the second row). The vertical line in each density plot is the estimated local mean from Split 2.

estimates indicate how important each BCUT is in the active and inactive classes. The larger values mean the corresponding BCUT descriptors are important for the discriminant analysis. For instance, using 0.15 as a threshold, BCUT3, BCUT4, BCUT5, and BCUT6 in the inactive class may distinguish inactive clusters from active clusters along the four descriptor dimensions. Wang (2005, Chapter 6) found that BCUT4 and BCUT6 are important variables. She used a subset K-nearest neighbour technique to identify the important variables.

Selecting important variables is an important topic. Since our focus here is to

explore multiple-mechanisms of drug data sets and subsets of descriptors, selecting important variables has not been carefully and systematically studied. In the future, we will focus on how to select important variables.

4.10 The CMDA Second Order Model (CMDA2)

We also consider the CMDA Second Order model (CMDA2), which is based on two-dimensional subspaces of descriptors. In the CMDA2 model, two variables can be simultaneously discriminate between classes. That is, activity is determined by interactions between two predictors. The CMDA2 model provides flexibility to identify this type of pattern. In this section, we will only present the EM derivations for the CMDA2 models, and one simulation example. The CMDA2 model will be applied to the NCI data later in Section 5.6.

This model explores the two-dimensional subsets of descriptors. For P descriptors, there are $P(P-1)/2$ components in each class, as each component is specified by a pair of descriptors. The second order model in the normal case can be written as

$$f(\mathbf{x}; \boldsymbol{\Psi_k}, \boldsymbol{\Psi_G}|y=k) = \sum_{j=1}^{P(P-1)/2} \pi_{jk} MVN(\mathbf{x_j}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l), \quad (4.26)$$

where $\boldsymbol{\Psi_k}$ represents local parameter for class k and $\boldsymbol{\Psi_G}$ is global parameter. A difference from the notation used in the CMDA1 model is that here j indexes a pair of descriptors, i.e. $j = \{1, 2, \ldots, P(P-1)/2\}$ corresponds to pairs $\{(1,2),(1,3),\ldots,(P-1,P)\}$. We also note that the global terms ($N(x_l; \mu_l, \sigma_l)$ in (4.26)) remain univariate for reasons of parsimony.

As before, the z_{ijk}'s represent the memberships of observations. The complete-data log likelihood for the second order model is

$$l_c(\mathbf{\Psi}) = \sum_{k=1}^{K} \sum_{j=1}^{P(P-1)/2} \sum_{i \in C_k} z_{ijk} [\log \pi_{jk} + \log MVN(\mathbf{x_j}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}) + \sum_{l \neq j} \log N(x_l; \mu_l, \sigma_l)], \tag{4.27}$$

where,

$$\log MVN(\mathbf{x_j}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}) = -\log 2\pi - \frac{1}{2} \log det(\Sigma_{jk}) - \frac{1}{2} (\mathbf{x_j} - \boldsymbol{\mu_{jk}})^T \boldsymbol{\Sigma_{jk}^{-1}} (\mathbf{x_j} - \boldsymbol{\mu_{jk}})$$

and

$$\log N(x_l; \mu_l, \sigma_l) = -\frac{1}{2} \log 2\pi - \frac{1}{2} \log(\sigma_l^2) - \frac{(x_l - \mu_l)^2}{2\sigma_l^2}.$$

Therefore the estimates in EM framework are:

E-Step:

$$\hat{z}_{ijk} = \frac{\hat{\pi}_{jk} MVN(\mathbf{x_j}; \mu_{\mathbf{jk}}, \boldsymbol{\Sigma_{jk}}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l)}{\sum_{j^*=1}^{p(p-1)/2} \hat{\pi}_{j^*k} MVN(\mathbf{x_{j^*}}; \boldsymbol{\mu_{j^*k}}, \boldsymbol{\Sigma_{j^*k}}) \prod_{l \neq j^*} N(x_l; \mu_l, \sigma_l)}; \tag{4.28}$$

M-Step:

$$\hat{\pi}_{jk} = \frac{\sum_{i \in C_k} \hat{z}_{ijk}}{\sum_{l=1}^{P(P-1)/2} \sum_{i \in C_k} \hat{z}_{ilk}}; \tag{4.29}$$

$$\boldsymbol{\hat{\mu}_{jk}} = \frac{\sum_{i \in C_k} \hat{z}_{ijk} \mathbf{x_{ij}}}{\sum_{i \in C_k} \hat{z}_{ijk}}; \tag{4.30}$$

$$\boldsymbol{\hat{\Sigma}_{jk}} = \frac{\sum_{i \in C_k} \hat{z}_{ijk} (\mathbf{x_{ij}} - \boldsymbol{\hat{\mu}_{jk}})(\mathbf{x_{ij}} - \boldsymbol{\hat{\mu}_{jk}})^T}{\sum_{i \in C_k} \hat{z}_{ijk}}; \tag{4.31}$$

$$\hat{\mu}_l = \frac{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk} x_{il}}{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk}}; \tag{4.32}$$

$$\hat{\sigma}_l^2 = \frac{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk} (x_{il} - \hat{\mu}_l)^2}{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk}}. \tag{4.33}$$

4.10.1 A Simulation Study

The simulation for the CMDA2 model is as follows:

- The simplest CMDA2 model is considered, $P = 3$, which means the data are 3-dimensional, and the CMDA2 model has 3 components. For class k, the 3-dimensional CMDA2 model in normal densities is written as

$$\begin{aligned} f(\mathbf{x}; \boldsymbol{\Psi_k}, \boldsymbol{\Psi_G}|y = k) &= \pi_{1k} MVN((x_1, x_2); \boldsymbol{\mu_{1k}}, \boldsymbol{\Sigma_{1k}}) N(x_3; \mu_3, \sigma_3) \\ &+ \pi_{2k} MVN((x_1, x_3); \boldsymbol{\mu_{2k}}, \boldsymbol{\Sigma_{2k}}) N(x_2; \mu_2, \sigma_2) \\ &+ \pi_{3k} MVN((x_2, x_3); \boldsymbol{\mu_{3k}}, \boldsymbol{\Sigma_{3k}}) N(x_1; \mu_1, \sigma_1). \quad (4.34) \end{aligned}$$

- Each element of the class specific means, $\boldsymbol{\mu_{jk}}$'s, is independently sampled from $U(-10, 10)$.

- Global means are independently sampled from $U(-10, 10)$.

- Class specific covariance matrices, $\boldsymbol{\Sigma_{jk}}$'s, are assumed to have equal off-diagonal covariance (e.g. $\sigma_{x_1x_2}$) that is 0.5. Diagonal entries (e.g. $\sigma^2_{x_1}$ and $\sigma^2_{x_2}$) for each class specific covariance matrix are independently sampled from $U(0.01, 2)$. These diagonal and off-diagonal entries have to satisfy $0.5 = \sigma_{x_1x_2} \leq \sigma_{x_1}\sigma_{x_2}$, since $|cov(X_1, X_2)| \leq \sigma_{X_1}\sigma_{X_2}$. Simulated parameter values that do not satisfy this constraint are discarded.

- The global variances are independently sampled from $U(0.01, 2)$.

- The training sample size is 360, while the testing size is 3600.

- The active:inactive balance is $1 : 9$.

	Ave. AHR	Se
Bayes	99.00	0.23
CMDA2	98.34	0.55
MclustDA	70.06	0.86

Table 4.19: Average AHR (%) and standard errors (%) calculated for Bayes, CMDA2 and MclustDA for the simulated data.

- Holding the parameters sampled fixed, 200 runs are implemented to get the average of parameter estimates and standard errors.

Since the data are unbalanced, AHR is used as comparison measurement between the CMDA2 model and the MclustDA model when the data are simulated from the CMDA2 model. The average parameter estimates and corresponding standard errors over 200 runs are summarized in Table 4.19. A hypothesis test indicates that CMDA2 has a significantly higher AHR than MclustDA.

Our simulation results show the CMDA2 model performs better than the MclustDA model, but further research needs to be done in order to explore the properties of the CMDA2 model.

We also apply the CMDA2 model to the NCI Antiviral AIDS data. First, however, the algorithm always converges to the degenerate solutions for the unconstrained CMDA2 model and secondly, it is very difficult to identify good local maxima even after the degenerate solutions are removed for a type of CMDA2 model with diagonal covariance matrices. Hence, a penalized CMDA2 model will be discussed in Section 5.6.

4.11 Discussion and Conclusion

In this chapter, a new mixture discriminant analysis model (CMDA) is introduced and discussed. The primary goal in drug discovery is to identify active compounds. However, the rareness of active compounds makes this difficult. A method like LDA, which assumes equal covariance matrices within each class is not very flexible. Approaches like QDA and MclustDA that assume different covariance matrices for different classes could suffer from the poor estimates of parameters for the active class due to the fewer active compounds in the data. The CMDA model is more flexible than LDA as different covariance matrices are assumed in each class, and more parsimonious than QDA and MclustDA as covariance matrices share some global parameters.

Comparisons between the CMDA first order model and the MclustDA model are conducted in two carefully designed scenarios: the true model is the CMDA1 model and the true model is the MclustDA model. The CMDA1 model outperforms the MclustDA model when the simulated data is generated from the CMDA1 model. When the true model is the MclustDA model, the MclustDA model is superior in low-dimensional cases and the CMDA1 model is superior in the high-dimensional cases, especially when the data are imbalanced.

In order to handle the degeneracy problem arising from estimating the parameters for the CMDA1 model, the Multi-step EM algorithm is designed to avoid degenerate solutions and converge to the best local optima. Compared to the EM algorithm, the Multi-step EM algorithm has significantly reduced the number of degenerate solutions. Since the Multi-step EM algorithm can not totally avoid

the degenerate solutions, however, a penalized and model-based approach will be discussed in the next chapter.

Chapter 5

Penalized Maximum Likelihood Estimation for the CMDA1 and CMDA2 models

In this chapter, we seek to use penalization as a way of avoiding degenerate solutions. Section 5.1 gives the notation used in this Chapter. The definition and proof of identifiability for the CMDA1 model is given in Section 5.2. The proof of asymptotic consistency of the penalized maximum likelihood estimate (PMLE) for the CMDA1 model is presented in Section 5.3. A simulation study using two penalty functions is given in Section 5.4 and the application of the PMLE on a drug data set is shown in Section 5.5. The PMLE approach is extended to the CMDA2 model in Section 5.6, and illustrated on the NCI data. Finally, conclusions are made in Section 5.7.

5.1 Introduction

Before discussing penalized estimation, let us recall the CMDA1 model and some notation used in Chapter 4. In general, the CMDA1 model specifies the density of feature vector $\mathbf{x}$ given class k as:

$$f(\mathbf{x}; \boldsymbol{\Psi_k}, \boldsymbol{\Psi_G}|y = k) = \sum_{j=1}^{P} \pi_{jk} h(x_j; \bar{\boldsymbol{\Phi}}_{\boldsymbol{jk}}) \prod_{l \neq j} h(x_l; \bar{\boldsymbol{\Phi}}_{\boldsymbol{l}}) \tag{5.1}$$

where $\boldsymbol{\Psi_k} = (\pi_{1k}, \ldots, \pi_{(p-1)k}, \bar{\boldsymbol{\Phi}}_{\mathbf{1k}}, \ldots, \bar{\boldsymbol{\Phi}}_{\boldsymbol{Pk}})^T$ is the vector containing all the unknown parameters specified in this mixture model for class k. Denote the global parameters $\boldsymbol{\Psi_G} = (\bar{\boldsymbol{\Phi}}_{\mathbf{1}}, \ldots, \bar{\boldsymbol{\Phi}}_{\boldsymbol{P}})^T$. The full set of parameters for all K classes is $\boldsymbol{\Psi} = (\boldsymbol{\Psi_1}, \ldots, \boldsymbol{\Psi_K}, \boldsymbol{\Psi_G})^T$. As in Chapter 4, we assume $h(\cdot)$ is a univariate normal density.

Some notation used in the model is as follows:

- The observations are represented by $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, where $\mathbf{x_i} \in \Re^P$ and y_i is a categorical variable with values $1, \ldots, K$. K is the total number of classes.
- $k = 1, \ldots, K$ indexes the K classes.
- $j = 1, \ldots, P$ indexes the P components in each class.

5.2 Identifiability

The estimation of $\boldsymbol{\Psi}$ on the basis of the observations $(\mathbf{x}_i, y_i)$ is only meaningful if $\boldsymbol{\Psi}$ is identifiable. In general, a parametric family of densities $f(\mathbf{x}; \boldsymbol{\Psi})$ is identifiable

if distinct values of the parameter $\mathbf{\Psi}$ determine distinct members of the family of densities $f(\mathbf{x};\mathbf{\Psi}) : \mathbf{\Psi} \in \Omega$, where Ω is the specified parameter space; that is

$$f(\mathbf{x};\mathbf{\Psi}) = f(\mathbf{x};\mathbf{\Psi}^*), \tag{5.2}$$

if and only if

$$\mathbf{\Psi} = \mathbf{\Psi}^*.$$

However, for mixture models, the above definition of identifiability of $\mathbf{\Psi}$ is not suitable. For instance, suppose $f(\mathbf{x};\mathbf{\Psi})$ has two component densities, $f(\mathbf{x};\mathbf{\Phi_a})$ and $f(\mathbf{x};\mathbf{\Phi_b})$, that belong to the same parametric family. Then (5.2) will still hold when the component label a and b and corresponding mixture weights π_a and π_b are interchanged in $\mathbf{\Psi}$. That is, $\mathbf{\Psi}$ is not identifiable. Indeed, if all the g component densities belong to the same parametric family, then $f(\mathbf{x};\mathbf{\Psi})$ is invariant under the $g!$ permutations of the component labels in $\mathbf{\Psi}$.

Let

$$f(\mathbf{x};\mathbf{\Psi}) = \sum_{j=1}^{g} \pi_j f(\mathbf{x};\mathbf{\Phi_j})$$

and

$$f(\mathbf{x};\mathbf{\Psi^*}) = \sum_{j=1}^{g^*} \pi_j^* f(\mathbf{x};\mathbf{\Phi_j^*})$$

be any two members of a parametric family of mixture models. This parametric family is said to be identifiable for $\mathbf{\Psi} \in \Omega$ if

$$f(\mathbf{x};\mathbf{\Psi}) \equiv f(\mathbf{x};\mathbf{\Psi^*})$$

if and only if $g = g^*$ and the component labels are permuted so that

$$\pi_j = \pi_j^* \text{ and } f(\mathbf{x}; \mathbf{\Phi_j}) = f(\mathbf{x}; \mathbf{\Phi_j^*}) \ (j = 1, \ldots, g). \tag{5.3}$$

So, there are two distinct definitions of identifiability: the identifiability of $\mathbf{\Psi}$ and the identifiability of a parametric family. For the CMDA1 model, the interchanging of component labels will result in a different parametric density, so $\mathbf{\Psi}$ is identifiable in this model. Therefore, the identifiability of the CMDA1 model as a parametric family is our focus.

Identifiability for the CMDA1 model with K classes is defined in a slightly different way as we need to show that K conditional mixture distributions are jointly identifiable:

$$\begin{aligned} f(\mathbf{x}; \mathbf{\Psi_1}|y = 1) &\equiv f(\mathbf{x}; \mathbf{\Psi_1^*}|y = 1), \\ f(\mathbf{x}; \mathbf{\Psi_2}|y = 2) &\equiv f(\mathbf{x}; \mathbf{\Psi_2^*}|y = 2), \\ &\ldots \\ f(\mathbf{x}; \mathbf{\Psi_K}|y = K) &\equiv f(\mathbf{x}; \mathbf{\Psi_K^*}|y = K). \end{aligned} \tag{5.4}$$

if and only if $\mathbf{\Psi_1} = \mathbf{\Psi_1^*}, \mathbf{\Psi_2} = \mathbf{\Psi_2^*}, \ldots$ and $\mathbf{\Psi_K} = \mathbf{\Psi_K^*}$.

For the case of univariate mixtures, we first construct a family of 1-dimensional component densities from which univariate mixtures are to be formed, i.e.

$$\mathcal{F} = \{f(x; \mathbf{\Phi}); \mathbf{\Phi} \in \mathbf{R}^m, x \in \mathbf{R}\}, \tag{5.5}$$

where $f(x; \mathbf{\Phi})$ is a component density. Then the class of finite mixtures of $\mathcal{F}$ with

the appropriate class of density functions, $\mathcal{H}$, is defined by

$$\mathcal{H} = \{H(x) : H(x) = \sum_{j=1}^{k} c_j f(x; \mathbf{\Phi_j}),\ c_j > 0, \tag{5.6}$$
$$\sum_{j=1}^{k} c_j = 1,\ f(x; \mathbf{\Phi_j}) \in \mathcal{F},\ k = 1, 2, \ldots\}$$

Before proving the identifiability of the CMDA1 model with 2 classes, we list several theoretical results necessary for the proof.

Theorem 1 (Yakowitz & Spragins (1968)) *A necessary and sufficient condition that the class $\mathcal{H}$ of all finite mixtures of the family $\mathcal{F}$ of (5.5) be identifiable is that $\mathcal{F}$ be a linearly independent set over the field of real numbers* $\mathbf{R}$. ■

Corollary 1 (Yakowitz & Spragins (1968)) *A necessary and sufficient condition that the class $\mathcal{H}$ of all finite mixtures of the family $\mathcal{F}$ of (5.5) be identifiable is that the image of $\mathcal{F}$ under any vector isomorphism on $< \mathcal{F} >$ (the span of $\mathcal{F}$) be linearly independent in the image space.* ■

Corollary 2 (Teicher (1963)) *The class of all finite mixtures of univariate normal distributions are identifiable.*

***Proof**: Let (μ, σ^2) denote the mean and variance of a typical member of $\mathcal{F}$, and $\phi(z; \mu, \sigma) = \exp(\mu z + \frac{1}{2}\sigma^2 z^2)$ be the moment-generating functions of members of $\mathcal{F}$.*

According to Corollary 1, we want to prove that if $\mathcal{H}$ is identifiable, then

$$\sum_{j=1}^{k} c_j \exp(\mu_j z + \frac{1}{2}\sigma_j^2 z^2) \equiv 0, \tag{5.7}$$

must give $c_j = 0, j = 1, 2, \ldots, k$.

First, we order σ_j^2, $j = 1, \ldots, k$, *and choose the largest one, say,* $\sigma_{j^*}^2$. *The term having* $\sigma_{j^*}^2$ *dominates the left side of (5.7). We divide both sides of (5.7) by* $\exp(\mu_{j^*} z + \frac{1}{2}\sigma_{j^*}^2 z^2)$ *and let* $z \to \infty$, *then we obtain*

$$|c_{j^*}| = \lim_{z\to\infty} |\frac{1}{\exp(\mu_{j^*} z + \frac{1}{2}\sigma_{j^*}^2 z^2)}|| \sum_{j=1 \& j\neq j^*}^{k} c_j \exp(\mu_j z + \frac{1}{2}\sigma_j^2 z^2)| = 0.$$

Hence $c_{j^*} = 0$ *and we remove this term from the left side of equation (5.7).*

If the largest σ *is not unique, e.g.* $\sigma_{j_1} = \sigma_{j_2}$, *then we compare* μ_{j_1} *and* μ_{j_2}. *If* $\mu_{j_2} < \mu_{j_1}$, *the same technique as above will be used to get* $c_{j_1} = 0$. *If* $\mu_{j_1} = \mu_{j_2}$, *these two terms can be combined, and then we repeat a similar procedure: compare* σ*'s, compare* μ*'s, prove* $c_j = 0$ *for some* j *and remove the terms with zero coefficient from (5.7).*

Therefore, we prove that the class of all finite mixtures of univariate normal distributions are identifiable as all the coefficients c_j *of equation (5.7) are 0.* ■

The above proof is slightly different from the original proof given in Teicher (1963) as we use the moment generating function, which provides a more direct proof.

We now use two methods to prove the identifiability of CMDA1 model: the first proves the theorem in terms of the product of univariate normal distributions and the second uses multivariate normal distributions. The second proof is on the basis that the CMDA1 model can be viewed as a mixture of multivariate normal distributions with diagonal covariance matrices.

Theorem 2 (Identifiability of the CMDA1 model) *The CMDA1 model is identifiable under the assumption that the descriptors (i.e. the x's) are independent within component densities.*

***Proof**: This proof is similar to the proof of Corollary 2.*

First, we define the sets $\mathcal{F}$ and $\mathcal{H}$ for the CMDA1 model in class k.

$$\mathcal{F} = \{f(\mathbf{x}; \mathbf{\Psi}) = N(x_j; \mu_{jk}, \sigma_{jk}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l)\}, \tag{5.8}$$

$$\mathcal{H} = \{H(\mathbf{x}) = \sum_{j=1}^{J} c_j N(x_j; \mu_{jk}, \sigma_{jk}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l),\ c_j > 0, \sum_{j=1}^{J} c_j = 1, J = 1, 2, \ldots\} \tag{5.9}$$

Let $\phi(z; \mathbf{\Psi})$ be the moment-generating function of $f(\mathbf{x}; \mathbf{\Psi})$, then

$$\phi(z; \mathbf{\Psi}) = \exp\{(\mu_{jk} z_{jk} + \sum_{l \neq j} \mu_l z_l) + \frac{1}{2}(\sigma_{jk}^2 z_{jk}^2 + \sum_{l \neq j} \sigma_l^2 z_l^2)\} \tag{5.10}$$

If $\mathcal{H}$ is identifiable, then according to Corollary 1,

$$\sum_{j=1}^{J} c_j \exp\{(\mu_{jk} z_{jk} + \sum_{l \neq j} \mu_l z_l) + \frac{1}{2}(\sigma_{jk}^2 z_{jk}^2 + \sum_{l \neq j} \sigma_l^2 z_l^2)\} \equiv 0 \tag{5.11}$$

must give $c_1 = c_2 = \ldots = c_J = 0$.

The same technique as in Corollary 2 is employed here to prove all the coefficients are zero in equation (5.11). Find the largest σ_*^2, and divide both sides of equation (5.11) by the term with the largest σ_*^2. Then the coefficient of that term is zero. If the largest σ_*^2 is not unique, the other σ^2's in the terms with the largest σ_*^2 will be compared. If the σ^2's are also the same, then identify the largest of the

corresponding μ_*'s. If the μ_*'s compared are equal, they can be combined. The same process is repeated until all terms of (5.11) are eliminated.

Therefore, we prove that the CMDA1 model is identifiable as all the coefficients of equation (5.11) are zero.

5.3 Asymptotic Consistency of the PMLE for the CMDA1 Model

As discussed in Section 4.5, the likelihood function of mixture models is unbounded for any given sample size. Hence, the ordinary maximum likelihood estimators (MLE) of mixture models are not consistent. This suggests that MLE's of the CMDA1 model are not consistent either.

In order to solve this problem, researchers commonly consider estimates on constrained parameter spaces. For example, Redner (1981) proved that the maximum likelihood estimate of $\mathbf{\Psi}$ exists and is globally consistent in every compact subparameter space containing the true parameter $\mathbf{\Psi}_0$. Hathaway (1985) proposed to estimate $\mathbf{\Psi}$ by maximizing the likelihood function with a restricted parameter space defined by the following constraint:

$$\ominus_c = \{\mathbf{\Psi} : \min_{i,j} \frac{\sigma_i}{\sigma_j} \geq c > 0\}$$

for some constant c. Hathaway's constrained MLE

$$\hat{\mathbf{\Psi}}_n = \arg \max_{\mathbf{\Psi} \in \ominus_c} l_n(\mathbf{\Psi})$$

is shown to be strongly consistent provided that the true mixing distribution $\mathbf{\Psi}_0$ belongs to $\ominus_c$. Despite the elegant results of Redner (1981) and Hathaway (1985), these methods all suffer, at least theoretically, from the risk that the true mixing distribution $\mathbf{\Psi}_0$ may not satisfy the constraint imposed (Chen, Tan & Zhang 2007). Chen et al. (2007) suggested that the approach of adding a penalty term to the ordinary log-likelihood function can avoid the above concern. They defined the penalized log likelihood as

$$Pl_n(\mathbf{\Psi}) = l_n(\mathbf{\Psi}) + p_n(\mathbf{\Psi}) \tag{5.12}$$

so that $p_n(\mathbf{\Psi}) \to -\infty$ as $\min\{\sigma_j : j = 1, \ldots, p\} \to 0$. Then the estimate of $\mathbf{\Psi}$ is the penalized maximum likelihood estimator (PMLE)

$$\tilde{\mathbf{\Psi}}_n = \arg\max_{\mathbf{\Psi}} Pl_n(\mathbf{\Psi}). \tag{5.13}$$

In this section, we introduce a family of simple penalty functions on the variances (especially class-specific variances) in the CMDA1 model. We will prove that the PMLE of the two-dimensional CMDA1 model is asymptotically consistent. The proof in Section 5.3.3 is based on Chen et al. (2007), which proved asymptotic consistency for penalized univariate normal mixtures. The proofs in this chapter are for two-dimensional multivariate normal mixtures with diagonal covariance matrices. In the following sections, we first borrow two technical lemmas from Chen et al. (2007) to assess the number of observations falling in a small neighborhood of the location parameters. Then we prove the asymptotic consistency of $\tilde{\mathbf{\Psi}}_n$ for general two-dimensional multivariate normal mixtures with diagonal covariance matrices. Since the two-dimensional CMDA1 model is a special case of a two-dimensional

multivariate normal mixture with diagonal covariance matrices, it is easy to conclude that the two-dimensional CMDA1 model is also asymptotically consistent.

5.3.1 Technical Lemmas

Two lemmas are borrowed from Chen et al. (2007) to prove the asymptotic consistency of the PMLE of the CMDA1 model. Please note that these lemmas are used for one dimension, i.e. $P = 1$ and a single group of mixtures (i.e. there are no classes). Thus the data are $x_1, \ldots, x_n$. We first give some definitions and quantities. A $P-$component one-dimensional normal mixture model is written as

$$f(\mathbf{x}; \mathbf{\Psi}) = \sum_{j=1}^{P} \pi_j f(\mathbf{x}; \mathbf{\Phi}_j), \tag{5.14}$$

where the mixing distribution is $\mathbf{\Psi} = (\pi_1, \ldots, \pi_{P-1}; \mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_P)$.

The basic idea of PMLE is to counter the effect of observations close to those location parameters with small scale parameters. For this purpose, assessing the number of observations falling in a small neighborhood of the location parameters in $\mathbf{\Psi}$ is important. The lemmas will aid in such assessments.

We first define

$$\Omega_n(\sigma) = \sup_{\mu} \sum_{i=1}^{n} I(0 < x_i - \mu < -\sigma \log \sigma) \tag{5.15}$$

which is the supremum (or least upper bound) of the number of observations falling into the positive side of a small neighborhood of all possible μ's. We are only interested in $\Omega_n(\sigma)$ when σ is very small, so we can assume $-\sigma \log(\sigma) > 0$. The number of observations falling into the negative side of μ can be assessed in a similar

way. Let $F_n(x) = n^{-1}\sum_{i=1}^{n} I(x_i \leq x)$ be the empirical distribution function. Then we have

$$\Omega_n(\sigma) = n \sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)].$$

Let $F = E(F_n)$ be the true cumulative distribution function.

We now define two quantities

$$T = \max\{\sup_{x} f(x; \mathbf{\Psi}_0), 8\}, \text{ and } \delta_n(\sigma) = -T\sigma\log(\sigma) + n^{-1},$$

where $\mathbf{\Psi}_0$ is the true mixing distribution. T is either the highest point in the true density or the constant 8, which is chosen for the convenience of the proof.

The following lemma uses Bahadur's representation to give an order assessment of $n^{-1}\Omega(\sigma)$.

Lemma 5.3.1 (Chen et al. (2007)) *Under the finite normal mixture model assumption, as $n \to \infty$ and almost surely, we have:*

1. *For each given σ between $\frac{8}{nT}$ and $\exp(-2)$. We have*

$$\sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \leq 2\delta_n(\sigma); \tag{5.16}$$

2. *For each given σ between 0 and $\frac{8}{nT}$,*

$$\sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \leq 2(\log n)^2/n. \tag{5.17}$$

Chen et al. (2007) use the same proof as below. We reproduce this proof here since a similar strategy will later be used in Section 5.3.3.

Proof:

1. *Let* $\eta_0, \eta_1, \ldots, \eta_n$ *be some real numbers such that*

$$\eta_0 = -\infty;\ F(\eta_i) = i/n,\ i = 1, \ldots, n-1;\ \eta_n = \infty.$$

We have

$$\begin{aligned} & \sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \\ \leq\ & \max_j[F_n(\eta_j - \sigma\log\sigma) - F_n(\eta_{j-1})] \\ \leq\ & \max_j[\{F_n(\eta_j - \sigma\log\sigma) - F_n(\eta_{j-1})\} - \{F(\eta_j - \sigma\log\sigma) - F(\eta_{j-1})\}] \\ & + \max_j[F(\eta_j - \sigma\log\sigma) - F(\eta_{j-1})]. \end{aligned} \tag{5.18}$$

By the mean value theorem and for some $\eta_j \leq \xi_j \leq \eta_j - \sigma\log\sigma$*, we have*

$$\begin{aligned} F(\eta_j - \sigma\log\sigma) - F(\eta_{j-1}) &= F(\eta_j - \sigma\log\sigma) - F(\eta_j) + n^{-1} \\ &= f(\xi_j; \mathbf{\Psi}_0)|\sigma\log\sigma| + n^{-1} \\ &\leq T|\sigma\log\sigma| + n^{-1} = \delta_n(\sigma). \end{aligned} \tag{5.19}$$

Then, we have $\max_j[F(\eta_j - \sigma\log\sigma) - F(\eta_{j-1})] \leq \delta_n(\sigma)$*. Further, for* $j = 1, \ldots, n$*, define*

$$\Delta_{nj} = |\{F_n(\eta_j - \sigma\log\sigma) - F_n(\eta_{j-1})\} - \{F(\eta_j - \sigma\log\sigma) - F(\eta_{j-1})\}|.$$

By the Bernstein inequality (Serfling, 1980), for any $t > 0$*, we have*

$$P\{\Delta_{nj} \geq t\} \leq 2\exp\{-\frac{n^2t^2}{2n\delta_n(\sigma) + \frac{2}{3}nt}\}. \tag{5.20}$$

Since $|\sigma\log\sigma|$ *is monotone in* σ*, for* $\exp(-2) > \sigma > 8/(nT)$*,*

$$|\sigma\log\sigma| \geq |\frac{8}{nT}\log\frac{8}{nT}| = \frac{8}{nT}\log\frac{nT}{8} \geq \frac{8\log n}{nT}.$$

By letting $t = \delta_n(\sigma)$ *in (5.20), we obtain*

$$\begin{aligned} P\{\Delta_{nj} \geq \delta_n(\sigma)\} &\leq 2\exp\{-\frac{3}{8}n\delta_n(\sigma)\} \\ &\leq 2\exp\{-\frac{3}{8}Tn|\sigma\log\sigma|\} \\ &\leq 2n^{-3}. \end{aligned} \tag{5.21}$$

Thus for any σ *in this range,*

$$P\{\max_j \Delta_{nj} \geq \delta_n(\sigma)\} \leq \sum_{j=1}^{n} P\{\Delta_{nj} \geq \delta_n(\sigma)\} \leq 2n^{-2}. \tag{5.22}$$

Linking the above inequality back to $\sup_\mu[F_n(\mu - \sigma\log\sigma) - F_n(\mu)]$ *(5.18), we get*

$$P\{\sup_\mu[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \geq 2\delta_n(\sigma)\} \leq P\{\max_j \Delta_{nj} \geq \delta_n(\sigma)\} \leq 2n^{-2}. \tag{5.23}$$

Then according to the Borel-Cantelli Lemma, $\sup_\mu[F_n(\mu-\sigma\log\sigma)-F_n(\mu)] \geq 2\delta_n(\sigma)$ *infinitely does not exist. So*

$$\sup_\mu[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \leq 2\delta_n(\sigma). \tag{5.24}$$

Hence we have proven the first part of the Lemma.

2. *When* $0 < \sigma < \frac{8}{nT}$*, by using the Bernstein inequality again, we have*

$$P\{\Delta_{nj} \geq t\} \leq 2\exp\{-\frac{n^2t^2}{2n\delta_n(\sigma) + \frac{2}{3}nt}\}.$$

Let $t = n^{-1}(\log\sigma)^2$

$$P\{\Delta_{nj} \geq n^{-1}(\log\sigma)^2\} \leq 2\exp\{-(\log\sigma)^2\} \leq n^{-3}.$$

Thus for any σ *in this range,*

$$P\{\max_j \Delta_{nj} \geq n^{-1}(\log\sigma)^2\} \leq \sum_{j=1}^{n} P\{\Delta_{nj} \geq n^{-1}(\log\sigma)^2\} \leq n^{-2}, \tag{5.25}$$

So

$$P\{\sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \geq 2n^{-1}(\log\sigma)^2\} \leq P\{\max_{j}\Delta_{nj} \geq n^{-1}(\log\sigma)^2\} \leq 2n^{-2}. \tag{5.26}$$

Then according to the Borel-Cantelli Lemma, $\sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \geq 2n^{-1}(\log\sigma)^2$ *does not infinitely exist. So* $\sup_{\mu}[F_n(\mu-\sigma\log\sigma)-F_n(\mu)] \leq 2n^{-1}(\log\sigma)^2$*, i.e. we have proven the second part of Lemma 5.3.1.* ■

The following Lemma strengthens the conclusions in Lemma 5.3.1, as the bounds can be violated by a zero-probability event for each σ and the union of zero-probability events may have non-zero probability as there are uncountable σ in the range. Here, we list Lemma 5.3.2 without proof. A proof is given in Chen et al. (2007).

Lemma 5.3.2 (Chen et al. (2007)) *Except for a zero-probability event not depending on* σ*, and under the same normal mixture assumptions, we have, for all large enough* n*,*

1. *For each given* σ*, which satisfies* $\frac{8}{nT} < \sigma < \exp(-2)$*. We have*

$$\sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \leq 4\delta_n(\sigma); \tag{5.27}$$

2. *Uniformly for* σ*,* $0 < \sigma < \frac{8}{nT}$*,*

$$\sup_{\mu}[F_n(\mu - \sigma\log\sigma) - F_n(\mu)] \leq 2(\log n)^2/n. \tag{5.28}$$

■

5.3.2 Penalized Likelihood and Penalty Functions

The penalized likelihood-based method is used to counter the unboundedness of $l_n(\mathbf{\Psi})$ while keeping the parameter space $\ominus$ unaltered. An important consideration is what kind of penalty function $p_n(\mathbf{\Psi})$ is eligible. Ridolfi & Idier (1999, 2000) proposed a class of penalty functions based on a Bayesian conjugate prior distribution, but the asymptotic properties of the corresponding PMLE were not discussed. Under some conditions on $p_n(\mathbf{\Psi})$, Ciuperca, Ridolfi & Idier (2003) attempted a proof of strong consistency of the PMLE of $\mathbf{\Psi}$ under the normal mixture model. Chen et al. (2007) noted that the proof contains a few loose steps which are difficult to tighten. Chen et al. (2007) employed a very different tactic in establishing the strong consistency of the PMLE for a class of penalty functions. In addition, they had shown that PMLE is asymptotically efficient.

In the thesis, we employ techniques of Chen et al. (2007) and expand their approach to the proof of the asymptotic consistency of PMLE for the CMDA1 model.

The penalty function $p_n(\mathbf{\Psi})$ in (5.12) for the CMDA1 model must satisfy the following conditions:

C1. $p_n(\mathbf{\Psi}) = \sum_j^P [\sum_{k=1}^K \tilde{p}_n(\sigma_{jk}) + \tilde{p}_n(\sigma_j)]$, where σ_{jk}'s are the class specific variances and σ_j's are the global variances;

C2. $\tilde{p}_n(\sigma) \to -\infty$ as $\sigma \to 0$;

C3. $\sup_{\sigma>0} \max\{0, \tilde{p}_n(\sigma)\} = o(n)$, and $\tilde{p}_n(\sigma) = o(n)$ at any fixed $\sigma > 0$.

C4. $\tilde{p}_n(\sigma) \leq 4(\log n)^2 \log \sigma$, when $\sigma \leq 8/(nT)$ as n is large enough.

In C1, penalty function $p_n(\mathbf{\Psi})$ is expressed as a sum of univariate penalty functions, a form that is convenient for numerical computation by the EM algorithm. Although each $\tilde{p}_n(\cdot)$ in $p_n(\mathbf{\Psi})$ could use a different penalty, for notational convenience we assume all penalties are of a single form $\tilde{p}_n(\sigma)$. To counter the effect of an unbounded density function of the CMDA1 model as $\sigma \to 0$, we must have $p_n(\mathbf{\Psi}) \to -\infty$ as $\sigma_j \to 0$ or $\sigma_{jk} \to 0$ for each $j = 1, \ldots, P$ and $k = 1, \ldots, K$. C3 rules out functions that substantially elevate or depress the penalized likelihood at any parameter value. At the same time, C3 allows the penalty to be very severe in a shrinking neighborhood of $\sigma = 0$. C4 determines the growth rate of penalty functions, insuring that the penalized log-likelihood can not be infinite for any given sample size. These four conditions are flexible and functions satisfying these conditions can be easily found and constructed. Some examples will be given in Section 5.4.

5.3.3 Asymptotic Consistency of the PMLE for Two-Dimensional Multivariate Mixture Models with Diagonal Covariance Matrices

We first prove the asymptotic consistency of the PMLE for two-dimensional multivariate normal mixtures, with diagonal covariance matrices and two mixture components. Since each class density of the two-dimensional CMDA1 model is a constrained case of multivariate normal mixtures, the theorem directly is applicable to the two-dimensional CMDA1 model.

We denote the two-dimensional multivariate normal mixture model as:

$$f(\mathbf{x};\mathbf{\Psi}) = \frac{\pi}{\sigma_1\sigma_2}\phi(\frac{x_1-\mu_1}{\sigma_1})\phi(\frac{x_2-\mu_2}{\sigma_2}) + \frac{1-\pi}{\delta_1\delta_2}\phi(\frac{x_1-\nu_1}{\delta_1})\phi(\frac{x_2-\nu_2}{\delta_2}), \tag{5.29}$$

which is based on some conditions: $\pi \neq 0, 1$ and $(\mu_1, \mu_2, \sigma_1, \sigma_2) \neq (\nu_1, \nu_2, \delta_1, \delta_2)$.

Hence we rewrite the log likelihood

$$l_n(\mathbf{\Psi}) = \sum_i \log\{\frac{\pi}{\sigma_1\sigma_2}\phi(\frac{x_{i1}-\mu_1}{\sigma_1})\phi(\frac{x_{i2}-\mu_2}{\sigma_2}) + \frac{1-\pi}{\delta_1\delta_2}\phi(\frac{x_{i1}-\nu_1}{\delta_1})\phi(\frac{x_{i2}-\nu_2}{\delta_2})\}, \tag{5.30}$$

and the penalty function

$$p_n(\mathbf{\Psi}) = \tilde{p}_n(\sigma_1) + \tilde{p}_n(\sigma_2) + \tilde{p}_n(\delta_1) + \tilde{p}_n(\delta_2), \tag{5.31}$$

where $\mathbf{\Psi} = \{\pi, \mu_1, \mu_2, \nu_1, \nu_2, \sigma_1, \sigma_2, \delta_1, \delta_2\}$. We penalize only variances. Here, the penalty function is a little different from that in Section 5.3.2 as there are no global parameters (σ_j's) in this unconstrained model. C2, C3 and C4 still hold for this penalty function, and C1 is rewritten as C1*:

C1*. $p_n(\mathbf{\Psi}) = \sum_{l=1}^{2}\tilde{p}_n(\sigma_l) + \sum_{l=1}^{2}\tilde{p}_n(\delta_l)$;

We also define the penalized log-likelihood function as

$$Pl_n(\mathbf{\Psi}) = l_n(\mathbf{\Psi}) + p_n(\mathbf{\Psi}). \tag{5.32}$$

We partition the parameter space into three regions. Figure 5.1 illustrates the partition of the parameter space in the variance component.

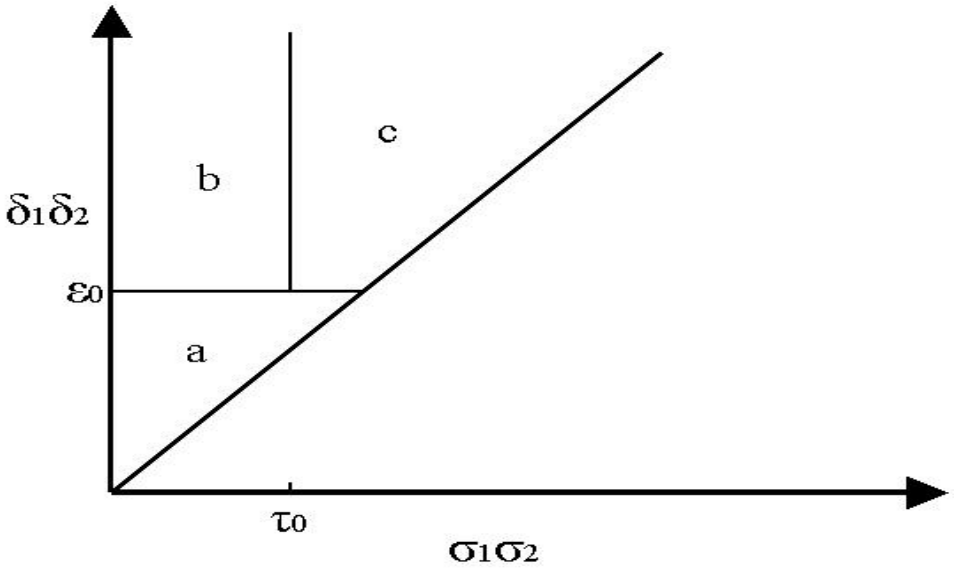

Figure 5.1: Partition of the parameter space Γ.

First, we define some constants that will be used later in the proof. Let $K_0 = E_0 \log f(\mathbf{X}; \mathbf{\Psi})$, where $E_0(\cdot)$ means expectation with respect to the true density $f(\mathbf{X}; \mathbf{\Psi}_0)$, i.e. $K_0 = \int \log f(\mathbf{X}; \mathbf{\Psi_0}) f(\mathbf{X}; \mathbf{\Psi_0}) d\mathbf{x}$. It is seen that $|K_0| < \infty$. Also we redefine the constant T in Section 5.3.1 as $T = \max\{\sup_{\mathbf{x}} f(\mathbf{x}; \mathbf{\Psi}_0), 8\}$, i.e. $f(\mathbf{x}; \mathbf{\Psi}_0)$ now is a mixture of multivariate normal distributions. Let ϵ_0 be a small positive constant such that

- $0 < \epsilon_0 < \exp(-2)$,
- $16T\epsilon_0(\log \epsilon_0)^2 \leq 1$,
- $-\log \epsilon_0 - (\log \epsilon_0)^2/2 \leq 2K_0 - 4$,

It is easy to see that as ϵ_0 goes to 0, the inequalities are satisfied. Hence, the existence of ϵ_0 is assured. For some small τ_0, we define three regions for the parameter

space in Figure 5.1

$$\begin{aligned}\Gamma_a &= \{\mathbf{\Psi} : \sigma_1\sigma_2 < \delta_1\delta_2 < \epsilon_0\},\\ \Gamma_b &= \{\mathbf{\Psi} : \sigma_1\sigma_2 \leq \tau_0, \delta_1\delta_2 \geq \epsilon_0\},\\ \Gamma_c &= \Gamma - (\Gamma_a \cup \Gamma_b).\end{aligned}$$

The exact size of τ_0 will be specified in the proof of Theorem 4. The three regions represent three situations. In Γ_a, the mixing distributions have at least one of the scale parameters along each dimension (X_1 and X_2) close to zero. In this case, the observations near any small location parameters contribute significantly to the log likelihood $l_n(\mathbf{\Psi})$, but will be countered by the penalty. The log likelihood contributions of the other observations can not exceed the likelihood at the true mixing distribution. Hence, the PMLE of $\mathbf{\Psi}$ has diminishing probability to be in Γ_a. In the second situation, at least one of the two mixing component distributions has one element of the scale parameter vector along X_1 dimension close to 0. When the mixing distribution has some of the scale parameters close to zero, the likelihood has two major sources: the observations near location parameters with corresponding small scale parameters, and the remaining observations. The first source is countered by the penalty. We will show that the likelihood from the second source is not large enough to exceed the likelihood at the true mixing distribution. Hence, the PMLE of $\mathbf{\Psi}$ also has diminishing probability to be in Γ_b. Once the possibility of the first two regions is eliminated, the consistency for the PMLE of $\mathbf{\Psi}$ in the third scenario Γ_c is established via the application of Kiefer & Wolfowitz (1956)'s consistency proof.

Theorems 3 and 4 will prove that asymptotically the PMLE cannot fall in the

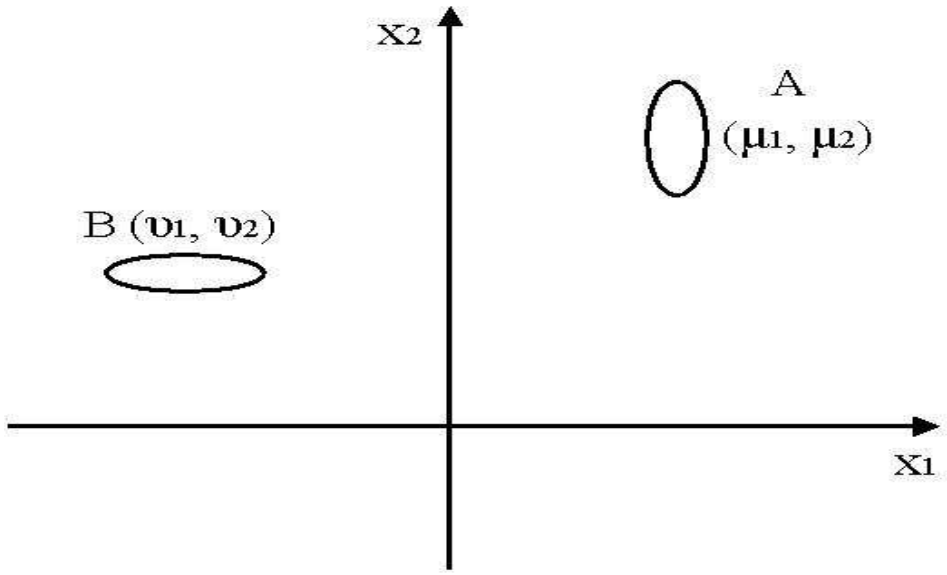

Figure 5.2: Defined regions A and B. The centre of the region A is (μ_1, μ_2), and the centre of the region B is (ν_1, ν_2).

first two regions. Theorem 5 shows that the PMLE must fall in the final region, Γ_c, and is thus consistent.

Theorem 3 (Γ_a)*: Under the assumptions that the data are a random sample from the model (5.29), and with* $Pl_n(\mathbf{\Psi})$ *defined as in (5.32), we have that almost surely when* $n \to \infty$*,*

$$\sup_{\mathbf{\Psi} \in \Gamma_a} Pl_n(\mathbf{\Psi}) - Pl_n(\mathbf{\Psi}_0) \to -\infty.$$

Proof: *Let* $\sigma_{min} = \min\{\sigma_1, \sigma_2\}$ *and* $A = \{i : (\frac{x_{i1}-\mu_1}{\sigma_1})^2 + (\frac{x_{i2}-\mu_2}{\sigma_2})^2 \leq \log^2(\sigma_{min})\}$ *and similarly* $B = \{i : (\frac{x_{i1}-\nu_1}{\delta_1})^2 + (\frac{x_{i2}-\nu_2}{\delta_2})^2 \leq \log^2(\delta_{min})\}$*. Figure 5.2 illustrates the regions of* (x_{i1}, x_{i2}) *for* $i \in A$ *and* $i \in B$ *in a two-dimensional space.*

For any index set, say S, we define,

$$l_n(\mathbf{\Psi}; S) = \sum_{i \in S} \log\{\frac{\pi}{\sigma_1\sigma_2}\phi(\frac{x_{i1}-\mu_1}{\sigma_1})\phi(\frac{x_{i2}-\mu_2}{\sigma_2}) + \frac{1-\pi}{\delta_1\delta_2}\phi(\frac{x_{i1}-\nu_1}{\delta_1})\phi(\frac{x_{i2}-\nu_2}{\delta_2})\}.$$

So $l_n(\mathbf{\Psi}) = l_n(\mathbf{\Psi}; A) + l_n(\mathbf{\Psi}; A^cB) + l_n(\mathbf{\Psi}; A^cB^c)$. We investigate the asymptotic order of these three terms. Let $n(A)$ be the number of observations in set A. From the fact that the mixture density is no larger than $\frac{1}{\sigma_1\sigma_2}$, we have

$$l_n(\mathbf{\Psi}; A) \leq -n(A)\log(\sigma_1\sigma_2), \tag{5.33}$$

and

$$l_n(\mathbf{\Psi}; A^cB) \leq -n(A^cB)\log(\delta_1\delta_2) \leq -n(B)\log(\delta_1\delta_2). \tag{5.34}$$

Let $A_1^* = \{i : (\frac{x_{i1}-\mu_1}{\sigma_1})^2 \leq \log^2(\sigma_{min})\}$ and $A_2^* = \{i : (\frac{x_{i2}-\mu_2}{\sigma_2})^2 \leq \log^2(\sigma_{min})\}$ respectively. It is clear that $A \subset A_1^*$ and $A \subset A_2^*$. Hence, $n(A) \leq \min\{n(A_1^*), n(A_2^*)\}$.

By Lemma 5.3.2, except for a zero probability event, as $n \to \infty$, we have

$$n(A_1^*) \leq \begin{cases} -4(\log n)^2 & \text{if } 0 < \sigma_1 < \frac{8}{nT} \\ -8 + 8Tn\sigma_1 \log\sigma_1 & \text{if } \frac{8}{nT} < \sigma_1 < \epsilon_0, \end{cases}$$

and

$$n(A_2^*) \leq \begin{cases} -4(\log n)^2 & \text{if } 0 < \sigma_2 < \frac{8}{nT} \\ -8 + 8Tn\sigma_2 \log\sigma_2 & \text{if } \frac{8}{nT} < \sigma_2 < \epsilon_0. \end{cases}$$

The above two bounds imply

$$\begin{aligned} n(A) &\leq \min\{n(A_1^*), n(A_2^*)\} \\ &\leq \begin{cases} -4(\log n)^2 & \text{when } 0 < \sigma_{min} < \frac{8}{nT} \\ -8 + 8Tn\sigma_{min}\log\sigma_{min} & \text{when } \frac{8}{nT} < \sigma_{min} < \epsilon_0. \end{cases} \end{aligned} \tag{5.35}$$

Therefore

$$l_n(\mathbf{\Psi}; A) \leq \begin{cases} -4(\log n)^2 \log(\sigma_1\sigma_2) & \text{if } 0 < \sigma_{min} < \frac{8}{nT} \\ (-8 + 8Tn\sigma_{min} \log \sigma_{min}) \log(\sigma_1\sigma_2) & \text{if } \frac{8}{nT} < \sigma_{min} < \epsilon_0. \end{cases}$$

From the two above inequalities and the conditions on the penalty functions, we obtain that

$$l_n(\mathbf{\Psi}; A) + \tilde{p}_n(\sigma_1) + \tilde{p}_n(\sigma_2) < 0, \tag{5.36}$$

when $0 < \sigma_{min} < \frac{8}{nT}$. *Also, when* $\frac{8}{nT} < \sigma_{min} < \epsilon_0$, *based on the choice of* ϵ_0, *almost surely, we arrive at the following bound:*

$$\begin{aligned} l_n(\mathbf{\Psi}; A) + \tilde{p}_n(\sigma_1) + \tilde{p}_n(\sigma_2) &\leq (-8 + 8Tn\sigma_{min} \log \sigma_{min}) \log(\sigma_1\sigma_2) \\ &\leq 8Tn\epsilon_0(\log \epsilon_0)^2 + 9 \log n. \end{aligned} \tag{5.37}$$

Similarly, we have

$$l_n(\mathbf{\Psi}; A^c B) + \tilde{p}_n(\delta_1) + \tilde{p}_n(\delta_2) \leq 8Tn\epsilon_0(\log \epsilon_0)^2 + 9 \log n. \tag{5.38}$$

For the observations that fall outside of both A *and* B, *their likelihood contributions are bounded by*

$$\begin{aligned} & \log\{\frac{\pi}{2\sigma_1\sigma_2} \exp(-\frac{1}{2}(\log(\sigma_1\sigma_2))^2) + \frac{1-\pi}{2\delta_1\delta_2} \exp(-\frac{1}{2}(\log(\delta_1\delta_2))^2)\} \\ \leq\ & \log\{\frac{\pi}{2\epsilon_0} \exp(-\frac{1}{2}(\log \epsilon_0)^2) + \frac{1-\pi}{2\epsilon_0} \exp(-\frac{1}{2}(\log \epsilon)^2)\} \\ \leq\ & -\log \epsilon_0 - \frac{1}{2}(\log \epsilon_0)^2, \end{aligned} \tag{5.39}$$

which is negative. At the same time, we also have

$$n(A) + n(B) \leq 2(\log n)^2/n + 2(\log n)^2/n = 4(\log n)^2/n < \frac{n}{2},$$

for large n. This shows that $n(A) + n(B) \leq \frac{n}{2}$, which means there are always less than half of the observations falling near location parameters (μ_1, μ_2) and (ν_1, ν_2). Then, we get the third bound

$$l_n(\mathbf{\Psi}; A^c B^c) \leq \frac{n}{2}\{-\log \epsilon_0 - (\log \epsilon_0)^2/2\}. \tag{5.40}$$

Then, combining the three bounds (5.37), (5.38) and (5.40), and recalling the choice of ϵ_0, we conclude that when $\mathbf{\Psi} \in \Gamma_a$,

$$\begin{aligned}
& Pl_n(\mathbf{\Psi}) \\
= \;& l_n(\mathbf{\Psi}) + p_n(\mathbf{\Psi}) \\
= \;& [l_n(\mathbf{\Psi}; A) + \tilde{p}_n(\sigma_1) + \tilde{p}_n(\sigma_2)] + [l_n(\mathbf{\Psi}; A^c B) + \tilde{p}_n(\delta_1) + \tilde{p}_n(\delta_2)] + [l_n(\mathbf{\Psi}; A^c B^c)] \\
\leq \;& 16Tn\epsilon_0(\epsilon_0)^2 + 18\log n + \frac{n}{2}(2K_0 - 4) \\
\leq \;& n(K_0 - 1) + 18\log n, \;\; a.s..
\end{aligned} \tag{5.41}$$

At the same time, by the strong law of large numbers

$$n^{-1} Pl_n(\mathbf{\Psi}_0) \to K_0 \;\; \textit{almost surely.} \tag{5.42}$$

Hence,

$$\sup_{\mathbf{\Psi} \in \Gamma_a} \; Pl_n(\mathbf{\Psi}) - Pl_n(\mathbf{\Psi}_0) \leq -n + 18\log n \to -\infty \tag{5.43}$$

almost surely as $n \to \infty$. ■

Theorem 3 tells us that the PMLE can not be in Γ_a.

Now we move to the second scenario, which requires results for the second region Γ_b. Unlike Γ_a, we have an unbounded Γ_b. Our first step is to compactify it. Define

a distance on Γ_b by

$$\begin{aligned} d(\mathbf{\Psi}, \mathbf{\Psi}^{'}) &= \arctan|\pi - \pi^{'}| + \sum_{l=1}^{2} \arctan|\mu_l - \mu_l^{'}| + \sum_{l=1}^{2} \arctan|\sigma_l - \sigma_l^{'}| \\ &\quad + \sum_{l=1}^{2} \arctan|\nu_l - \nu_l^{'}| + \sum_{l=1}^{2} \arctan|\delta_l - \delta_l^{'}|. \end{aligned} \tag{5.44}$$

Under this distance, Γ_b is a totally bounded finite dimensional set, so it can be compactified. For convenience, we use the same notation Γ_b for the compact set. Define

$$\begin{aligned} g(\mathbf{x}; \mathbf{\Psi}) &= a_1 \frac{\pi}{2} \phi(\frac{x_{i1} - \mu_1}{\sqrt{2}\sigma_1}) \phi(\frac{x_{i2} - \mu_2}{\sqrt{2}\sigma_2}) \\ &\quad + a_2 \frac{1-\pi}{\delta_1 \delta_2} \phi(\frac{x_{i1} - \nu_1}{\delta_1}) \phi(\frac{x_{i2} - \nu_2}{\delta_2}). \end{aligned} \tag{5.45}$$

on Γ_b, with $a_1 = I(\mu_1 \neq \pm\infty, \sigma_1 \neq 0; \mu_2 \neq \pm\infty, \sigma_2 \neq 0)$ and $a_2 = I(\nu_1 \neq \pm\infty; \nu_2 \neq \pm\infty)$. The function $g(\cdot)$ is well defined over the entire space with some continuity. The changes in the normal densities of $g(\cdot)$ ensure that the integral of $g(\cdot)$ is no larger than 1 in order to use Jensen's inequality in the following proof.

Let $K(\mathbf{\Psi}) = E_0 \log g(\mathbf{X}; \mathbf{\Psi})$, which has the following property.

Lemma 5.3.3 *For any* $\{\mathbf{\Psi}_n, n = 1, 2, \ldots\} \subseteq \Gamma_b$, *such that* $\mathbf{\Psi}_n \to \mathbf{\Psi}$, *we have*

$$\overline{\lim}_{n\to\infty} K(\mathbf{\Psi}_n) \leq K(\mathbf{\Psi}). \tag{5.46}$$

Proof: *For any* $\rho > 0$, *define*

$$g(\mathbf{x}; \mathbf{\Psi}, \rho) = \sup\{g(\mathbf{x}; \mathbf{\Psi}^{'}); d(\mathbf{\Psi}, \mathbf{\Psi}^{'}) < \rho, \mathbf{\Psi}^{'} \in \Gamma_b\}.$$

Note that $\lim_{\rho\to 0} g(\mathbf{x}; \mathbf{\Psi}, \rho) = g(\mathbf{x}; \mathbf{\Psi})$, *and* $\sup\{g(\mathbf{x}; \mathbf{\Psi}); \mathbf{\Psi} \in \Gamma_b\} \leq \frac{1}{\epsilon_0}$.

By the dominated convergence theorem,

$$\lim_{\rho \to 0} E_0 \log g(\mathbf{x}; \mathbf{\Psi}, \rho) = E_0 \log g(\mathbf{x}; \mathbf{\Psi}) = K(\mathbf{\Psi}).$$

Let $\rho_n = d(\mathbf{\Psi}, \mathbf{\Psi}_n)$*, we have* $K(\mathbf{\Psi}_n) \leq E_0 \log g(\mathbf{x}; \mathbf{\Psi}, \rho_n)$*, therefore,*

$$\overline{\lim}_{n \to \infty} K(\mathbf{\Psi}_n) \leq \overline{\lim}_{n \to \infty} E_0 \log g(\mathbf{x}; \mathbf{\Psi}, \rho_n) = E_0 \log g(\mathbf{x}; \mathbf{\Psi}) = K(\mathbf{\Psi}).$$

■

Theorem 4 (Γ_b)*: Under the assumptions that the data are a random sample from the model (5.29), and with* $Pl_n(\mathbf{\Psi})$ *defined as in (5.32), we have that almost surely when* $n \to \infty$*,*

$$\sup_{\mathbf{\Psi} \in \Gamma_b} Pl_n(\mathbf{\Psi}) - Pl_n(\mathbf{\Psi}_0) \to -\infty.$$

Proof: *Lemma 5.3.3 tells us that there is* $\mathbf{\Psi}^* \in \Gamma_b$ *such that* $K^* = K(\mathbf{\Psi}^*) = \sup\{K(\mathbf{\Psi}) : \mathbf{\Psi} \in \Gamma_b\}$*. Let* $\delta = \delta(\tau_0) = -E_0 \log\{\frac{g(\mathbf{x}; \mathbf{\Psi}^*)}{f(\mathbf{x}; \mathbf{\Psi}_0)}\} = K_0 - K^*$*. The dependence of* $\delta(\tau_0)$ *on* τ_0 *is due to the dependence of* $\mathbf{\Psi}^*$ *on boundary* τ_0*.* $\delta(\tau_0)$ *is a decreasing function of* τ_0*.*

For small τ_0*, we have*

$$\begin{aligned} g(\mathbf{x}; \mathbf{\Psi}) \leq a_1 \frac{\pi}{2\sigma_1\sigma_2} \phi(\frac{x_{i1} - \mu_1}{\sqrt{2}\sigma_1}) \phi(\frac{x_{i2} - \mu_2}{\sqrt{2}\sigma_2}) \\ + a_2 \frac{1 - \pi}{\delta_1 \delta_2} \phi(\frac{x_{i1} - \nu_1}{\delta_1}) \phi(\frac{x_{i2} - \nu_2}{\delta_2}). \end{aligned} \tag{5.47}$$

By Jensen's inequality, we have $\delta(\tau_0) > 0$*. We can find* τ_0 *such that*

1. $\tau_0 < \epsilon_0$,
2. $8T\tau_0(\log \tau_0)^2 < 2\delta(\epsilon_0)/5 < 2\delta(\tau_0)/5$.

We now proceed to show that the PMLE can not be in Γ_b. *For any* $\mathbf{\Psi} \in \Gamma_b$

$$\lim_{\rho \to 0} E_0 \log g(\mathbf{x}; \mathbf{\Psi}, \rho) = E_0 \log g(\mathbf{x}; \mathbf{\Psi}) \leq K^*.$$

Hence for each $\mathbf{\Psi} \in \Gamma_b$, *there exists a* $\rho(\mathbf{\Psi}) > 0$, *such that*

$$E_0 \log g(\mathbf{x}; \mathbf{\Psi}, \rho(\mathbf{\Psi})) < K^* + \frac{\delta(\tau_0)}{10} = K_0 - \frac{9\delta(\tau_0)}{10}.$$

Let $B(\mathbf{\Psi}; \rho(\mathbf{\Psi})) = \{\mathbf{\Psi}^{'} \in \Gamma_b : d(\mathbf{\Psi}, \mathbf{\Psi}^{'}) < \rho(\mathbf{\Psi})\}$, *then* $B(\mathbf{\Psi}; \rho(\mathbf{\Psi}))$ *forms an open cover of* Γ_b. *From the compactness of* Γ_b, *there are a finite number of* $\mathbf{\Psi}_k, \rho_k$, *with* $k = 1, 2, \ldots, K$, *such that*

$$\bigcup_{k=1}^{K} B(\mathbf{\Psi}_k, \rho_k) = \Gamma_b.$$

Hence,

$$\sup\{\sum_{i=1}^{n} \log g(\mathbf{x_i}; \mathbf{\Psi}) : \mathbf{\Psi} \in \Gamma_b\} \leq \max_k \{\sum_{i=1}^{n} \log g(\mathbf{x_i}; \mathbf{\Psi}_k, \rho_k)\}.$$

For each k, *by the law of large numbers,*

$$\sum_{i=1}^{n} \log g(\mathbf{x_i}; \mathbf{\Psi}_k, \rho(\mathbf{\Psi}_k)) \leq n\{K_0 - \frac{9\delta(\tau_0)}{10} + o(1)\}.$$

Consequently

$$\sup\{\sum_{i=1}^{n} \log g(\mathbf{x_i}; \mathbf{\Psi}) : \mathbf{\Psi} \in \Gamma_b\} \leq n\{K_0 - \frac{9\delta(\tau_0)}{10} + o(1)\}.$$

The likelihood contribution of observations in A *is no larger than* $-\log(\sigma_1\sigma_2) + \log g(\mathbf{x}; \mathbf{\Psi})$. *For other observations, their likelihood contributions are less than* $\log g(\mathbf{x}; \mathbf{\Psi})$. *This is seen by the fact that when* $(\frac{x_{i1}-\mu_1}{\sigma_1})^2 + (\frac{x_{i2}-\mu_2}{\sigma_2})^2 \geq \log^2(\sigma_{min})$, *and* $\sigma_1\sigma_2$ *are sufficiently small,*

$$\begin{aligned} &\frac{1}{\sigma_1\sigma_2} \exp\{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}\} \exp\{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}\} \\ &\quad \leq \exp\{-\frac{(x_1-\mu_1)^2}{4\sigma_1^2}\} \exp\{-\frac{(x_2-\mu_2)^2}{4\sigma_2^2}\}. \end{aligned} \tag{5.48}$$

Hence, combined with the properties of the penalty function (C1-C4) and (5.48), we have

$$\begin{aligned}
&\sup_{\Gamma_b} Pl_n(\mathbf{\Psi}) - Pl_n(\mathbf{\Psi}_0) \\
\leq\ & \sup_{\sigma_1\sigma_2<\tau_0} \{\sum_{i\in A}(-\log(\sigma_1\sigma_2) + \tilde{p}_n(\sigma_1) + \tilde{p}_n(\sigma_2))\} + \sup_{\Gamma_b} \sum_{i=1}^{n} \log\{\frac{g(\mathbf{x_i};\mathbf{\Psi})}{f(\mathbf{x_i};\mathbf{\Psi}_0)}\} - p_n(\mathbf{\Psi}_0) \\
\leq\ & 8Tn\tau_0(\log\tau_0)^2 + 9\log n - \frac{9n\delta(\tau_0)}{10} - p_n(\mathbf{\Psi}_0) \\
\leq\ & -\frac{n\delta(\tau_0)}{10} + 9\log n - p_n(\mathbf{\Psi}_0) \to -\infty \ \textit{as}\ n\to\infty. \qquad (5.49)
\end{aligned}$$

■

We now claim the strong consistency of the PMLE.

Theorem 5 *Under the assumptions defined before, for any mixing distribution* $\mathbf{\Psi}_n = \mathbf{\Psi}_n(\mathbf{X_1},\ldots,\mathbf{X_n})$*, satisfying*

$$Pl_n(\mathbf{\Psi}_n) - Pl_n(\mathbf{\Psi}_0) > c > -\infty, \qquad (5.50)$$

we have that $\mathbf{\Psi}_n \to \mathbf{\Psi}_0$ *almost surely as* $n\to\infty$*.*

Proof: *By Theorem 3 and Theorem 4, with probability one,* $\mathbf{\Psi}_n \in \Gamma_c$ *as* $n\to\infty$*. Confining the mixing distribution* $\mathbf{\Psi}$ *in* Γ_c *is equivalent to placing a positive constant lower bound for the variance parameters. Thus, the consistency is covered by the result in Kiefer & Wolfowitz (1956). Note that their proof can be modified to accommodate a penalty of size* $o(n)$ *due to the conditions of their proof.* ■

Let $\hat{\mathbf{\Psi}}_n$ be the PMLE that maximizes $Pl_n(\mathbf{\Psi})$. By definition, $Pl_n(\hat{\mathbf{\Psi}}_n) - Pl_n(\mathbf{\Psi}_0) > 0$ and therefore $\hat{\mathbf{\Psi}}_n \to \mathbf{\Psi}_0$ almost surely. Hence we have proved that PMLE $\hat{\mathbf{\Psi}}_n$ is strongly consistent in the Γ_c space.

5.3.4 Asymptotic Consistency of PMLE for the CMDA1 Model in Two Dimensions

Now we can apply the above theorems to the two-dimensional CMDA1 model with two classes. We put penalties on both class-specific and global variances. The log-likelihood of this model can be written in terms of standard normal distributions

$$
\begin{aligned}
&l_n(\mathbf{\Psi}_{CMDA}) = \\
&\sum_{i\in C_1} \log\{\pi_1 \frac{1}{\sigma_{11}\sigma_2}\phi(\frac{x_{i1}-\mu_{11}}{\sigma_{11}})\phi(\frac{x_{i2}-\mu_2}{\sigma_2}) + (1-\pi_1)\frac{1}{\sigma_1\sigma_{12}}\phi(\frac{x_{i1}-\mu_1}{\sigma_1})\phi(\frac{x_{i2}-\mu_{12}}{\sigma_{12}})\} \\
&+ \sum_{i\in C_2} \log\{\pi_2 \frac{1}{\sigma_{21}\sigma_2}\phi(\frac{x_{i1}-\mu_{21}}{\sigma_{21}})\phi(\frac{x_{i2}-\mu_2}{\sigma_2}) + (1-\pi_2)\frac{1}{\sigma_1\sigma_{22}}\phi(\frac{x_{i1}-\mu_1}{\sigma_1})\phi(\frac{x_{i2}-\mu_{22}}{\sigma_{22}})\}
\end{aligned}
$$

and the penalty function as

$$p_n(\mathbf{\Psi}_{CMDA}) = \tilde{p}_n(\sigma_{11}) + \tilde{p}_n(\sigma_{12}) + \tilde{p}_n(\sigma_{21}) + \tilde{p}_n(\sigma_{22}) + \tilde{p}_n(\sigma_1) + \tilde{p}_n(\sigma_2). \quad (5.51)$$

In order to prove the asymptotic consistency of the PMLE, we divide the log-likelihood function into two parts, each of which represents one sample group (class 1 or class 2).

$$l_n(\mathbf{\Psi}_{CMDA}) = l_{n1}(\mathbf{\Psi}_1) + l_{n2}(\mathbf{\Psi}_2) \quad (5.52)$$

where

$$
\begin{aligned}
&l_{n1}(\mathbf{\Psi}_1) = \\
&\sum_{i\in C_1} \log\{\pi_1 \frac{1}{\sigma_{11}\sigma_2}\phi(\frac{x_{i1}-\mu_{11}}{\sigma_{11}})\phi(\frac{x_{i2}-\mu_2}{\sigma_2}) + (1-\pi_1)\frac{1}{\sigma_1\sigma_{12}}\phi(\frac{x_{i1}-\mu_1}{\sigma_1})\phi(\frac{x_{i2}-\mu_{12}}{\sigma_{12}})\}, \\
&l_{n2}(\mathbf{\Psi}_2) = \\
&\sum_{i\in C_2} \log\{\pi_2 \frac{1}{\sigma_{21}\sigma_2}\phi(\frac{x_{i1}-\mu_{21}}{\sigma_{21}})\phi(\frac{x_{i2}-\mu_2}{\sigma_2}) + (1-\pi_2)\frac{1}{\sigma_1\sigma_{22}}\phi(\frac{x_{i1}-\mu_1}{\sigma_1})\phi(\frac{x_{i2}-\mu_{22}}{\sigma_{22}})\}.
\end{aligned}
$$

and

$$\boldsymbol{\Psi}_1 = \{\mu_{11}, \mu_{12}, \sigma_{11}, \sigma_{12}, \mu_1, \mu_2, \sigma_1, \sigma_2\},$$

$$\boldsymbol{\Psi}_2 = \{\mu_{21}, \mu_{22}, \sigma_{21}, \sigma_{22}, \mu_1, \mu_2, \sigma_1, \sigma_2\}.$$

We define

$$Pl_{n1}(\boldsymbol{\Psi}_1) = l_{n1}(\boldsymbol{\Psi}_1) + p_{n1}(\boldsymbol{\Psi}_1), \tag{5.53}$$

$$Pl_{n2}(\boldsymbol{\Psi}_2) = l_{n2}(\boldsymbol{\Psi}_2) + p_{n2}(\boldsymbol{\Psi}_2). \tag{5.54}$$

Here, $p_{n1}(\boldsymbol{\Psi}_1)$ and $p_{n2}(\boldsymbol{\Psi}_2)$ are the penalty functions for the parameters in class 1 and class 2, respectively. Recall that the penalized MLE is the parameter value (multi-dimensional) that maximizes the penalized log-likelihood function. From the proof in Section 5.3.3, it is seen that the maximum point of both Pl_{n1} and Pl_{n2} are in the small neighborhood of the true parameter value. In fact, outside of any small neighborhood, the suprema of two penalized log-likelihoods are smaller than the penalized log-likelihoods at the true parameter value by an order of n. Thus, the maximum of the sum of these two penalized log-likelihood must be inside any small neighborhood of the true parameter value as $n \to \infty$. That is, the penalized maximum likelihood of the CMDA1 model is asymptotically consistent.

5.4 A Simulation Study Using Two Penalty Functions

We suggest two kinds of penalty functions, each of which comes from different perspectives. One theme in these penalty functions is that they are related to the

inverse gamma distribution, which is the conjugate prior for the complete data likelihood. Penalization by an inverse gamma distribution induces no structural changes in the EM algorithm and explicitness of the estimates is maintained.

The penalized log likelihood increases after each iteration (Green 1990). Furthermore, penalization does not increase the computational burden as Green (1990) pointed out that the penalized EM algorithm converges as least as quickly as the standard one.

Section 5.4.1 discusses a non-Bayesian penalty function, while Section 5.4.2 focuses on a penalty function from Bayesian framework. The EM derivations for two penalty functions will be presented before carrying out simulations. The simulations are similar to those in Section 4.7.1.

5.4.1 A Non-Bayesian Perspective

The first penalty function is motivated by Chen et al. (2007). Some modifications have been made in order to account the structure of the CMDA1 model. From the experiments in Chapter 4, we found that the global variances do not converge to degenerate solutions. Therefore, we only penalize local variances.

For the CMDA1 model, the penalty function is

$$P_1(\mathbf{\Psi}) = \sum_{k=1}^{K}\{-\frac{1}{n_k}\sum_{j=1}^{P}(\frac{D_{jk}S_{jk}}{\sigma_{jk}^2}) - \frac{1}{n_k}\sum_{j=1}^{P}\log(\frac{\sigma_{jk}^2}{S_{jk}})\}, \tag{5.55}$$

where n_k is the number of observations in class k; S_{jk} is the sample variance along x_j for the observations in the k^{th} class between the first and third sample quartiles and D_{jk} is an arbitrary positive constant. Here we need to emphasize that the

tuning parameter D_{jk} is not a regularized parameter, but a constant that adjusts the penalty functions. Since the within cluster variances are likely to be smaller than the marginal sample variance of x_j in class k, a trimmed variance S_{jk} is used. This will also reduce sensitivity to outliers.

A desirable property of MLEs is their invariance: The MLE of a parameter can be used to calculate the MLE of a one-to-one function of the parameter. PMLE's do not necessarily possess this functional invariance property. In order to incorporate this nice feature, in the PMLE, S_{jk}'s are introduced in the penalty function. In the following sections, the choice of D_{jk} will be discussed in detail. It is not hard to verify that this penalty function satisfies the four requirements (C1-C4) in Section 5.3.2.

EM Derivation of PMLE

The penalized complete data log-likelihood is

$$l_{pc}(\mathbf{\Psi}) = \sum_{k=1}^{K}\sum_{j=1}^{P}\Big\{\sum_{i\in C_k} z_{ijk}\Big[\log\pi_{jk} - \log\sigma_{jk} - \frac{(x_{ij}-\mu_{jk})^2}{2\sigma_{jk}^2} + \sum_{l\neq j}(-\log\sigma_l - \frac{(x_{il}-\mu_l)^2}{2\sigma_l^2})\Big] - \frac{1}{n_k}\frac{D_{jk}S_{jk}}{\sigma_{jk}^2} - \frac{1}{n_k}\log\frac{\sigma_{jk}^2}{S_{jk}}\Big\}. \tag{5.56}$$

The closed form expressions for parameter estimates in the EM framework at the $(a+1)^{th}$ iteration are:

E-Step:

$$\begin{aligned}\hat{z}_{ijk}^{(a+1)} &= \hat{p}(\text{the } i^{th} \text{ observation} \in \text{the } j^{th} \text{ component}|\text{the } k^{th} \text{ class}) \\ &= \frac{\hat{\pi}_{jk}^{(a)} N(x_{ij};\hat{\mu}_{jk}^{(a)},\hat{\sigma}_{jk}^{(a)})\prod_{l\neq j} N(x_{il};\hat{\mu}_l^{(a)},\hat{\sigma}_l^{(a)})}{\sum_{m=1}^{P}\hat{\pi}_{mk}^{(a)} N(x_{im};\hat{\mu}_{mk}^{(a)},\hat{\sigma}_{mk}^{(a)})\prod_{l\neq m} N(x_{il};\hat{\mu}_l^{(a)},\hat{\sigma}_l^{(a)})}\end{aligned} \tag{5.57}$$

M-Step:

$$\hat{\pi}_{jk}^{(a+1)} = \frac{\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}}{\sum_{j=1}^{p}\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}}, \tag{5.58}$$

$$\hat{\mu}_{jk}^{(a+1)} = \frac{\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)} x_{ij}}{\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}}, \tag{5.59}$$

$$\hat{\sigma}_{jk}^{2\,(a+1)} = \frac{\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}(x_{ij} - \hat{\mu}_{jk}^{(a)})^2 + 2D_{jk}S_{jk}/n_k}{\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)} + 2/n_k}, \tag{5.60}$$

$$\hat{\mu}_{l}^{(a+1)} = \frac{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)} x_{il}}{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}}, \tag{5.61}$$

$$\hat{\sigma}_{l}^{2(a+1)} = \frac{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}(x_{il} - \hat{\mu}_{l}^{(a)})^2}{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(a+1)}}. \tag{5.62}$$

We note that the only change is the inclusion of extra terms in the numerator and denominator of $\hat{\sigma}_{jk}^{2\,(a+1)}$ in (5.60).

Choice of D_{jk}

We consider using the same constant for each D_{jk} for convenience, i.e. $D_{11} = \ldots = D_{PK} = D$. D is an adjusting parameter and corresponds to the prior mode. First, a range of D values, $(0.001, 0.005, 0.01, 0.05, 0.1, 2, 3, 4, 5, 6)$ is used to test sensitivity to the tuning parameter D using the model from combination 7 in Table 4.4. The true parameter values for this model are given in Table 5.1. The five factors of the 7^{th} combination are: dimensionality is 2, the class variances are the same, the sample size is large, the data are unbalanced, and the means are well separated. The ratio of active and inactive compounds in this example is 1 : 9. The true value of σ_{11} is close to 0, i.e. $\sigma_{11} = 0.1639$. Figure 5.3 shows us that large D values push

	True Parameters			
μ_{jk}	$\mu_{11} = -4.4508$	$\mu_{12} = -5.7968$	$\mu_{21} = 2.4323$	$\mu_{22} = 1.5012$
μ_j	$\mu_1 = 4.4315$	$\mu_2 = 4.5122$		
σ_{jk}	$\sigma_{11} = 0.1639$	$\sigma_{12} = 0.1639$	$\sigma_{21} = 1.1383$	$\sigma_{22} = 1.1383$
σ_j	$\sigma_1 = 0.1709$	$\sigma_2 = 1.0361$		

Table 5.1: The model corresponding to combination 7 in Table 4.4.

the PMLE far away from the true σ_{11}, while the first several small D values have little effect on the PMLE.

One further experiment is conducted to verify if D is scale-invariant. We multiply x_1 by 1000, and repeat the above experiment. The similar boxplot of PMLEs of $\sigma_{11} = 0.1639$ obtained from the new experiment indicates that D does not depend on data and gives consistent estimates in the range $(0.001, 1)$. It is not surprising that D is independent from data as S_{jk} and σ_{jk}^2 in the penalty function (5.57) cancel the scaling effect. In the remainder of this chapter, $D = 0.1$ is chosen for the penalty function P_1.

Degenerate Solutions

As in Section 4.7.2, we can study whether the EM and modified EM algorithms (Multi-step EM) return degenerate PMLE's. Models from the first 16 combinations in Table 4.4 are used here to verify that the PMLE can avoid degenerate solutions. Combining the results from the previous chapter, Table 5.2 clearly shows that the PMLE totally avoids degenerate solutions no matter which algorithm is used. For combinations 17-32, the same conclusion can be made, i.e. PMLE does not converge

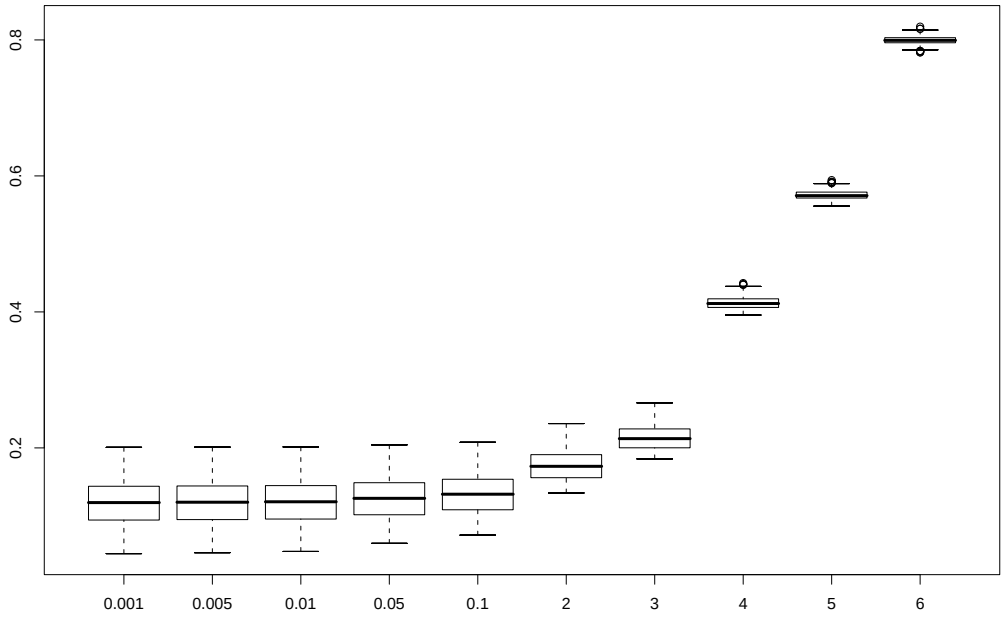

Figure 5.3: Boxplot of PMLEs of $\sigma_{11} = 0.1639$ vs. D (0.001, 0.005, 0.01, 0.05, 0.1, 2, 3, 4, 5, 6).

to degenerate solutions for high-dimensional cases.

Parameter Estimates

We report MSE's for estimates of class-specific variances ($\sigma_{ij}^2, i = 1, 2, j = 1, 2$) for the first 16 simulation models in Table 4.4. Two algorithms are used to calculate PMLE's and MLE's: the Multi-step EM algorithm and the EM algorithm. So there are four different scenarios: PMLE/Multi-step EM, PMLE/EM, MLE/Multi-step EM and MLE/EM.

MSE's of both PMLE and MLE calculated via the conventional EM algorithm

	1	2	3	4	5	6	7	8
PMLE (Multi-step EM)	0	0	0	0	0	0	0	0
MLE (Multi-step EM)	0	0	0	20	0	0	0	1
PMLE (EM)	0	0	0	0	0	0	0	0
MLE (EM)	103	200	174	200	3	200	44	199
	9	10	11	12	13	14	15	16
PMLE (Multi-step EM)	0	0	0	0	0	0	0	0
MLE (Multi-step EM)	0	1	0	14	0	0	0	1
PMLE (EM)	0	0	0	0	0	0	0	0
MLE (EM)	24	145	77	157	0	68	1	83

Table 5.2: Degenerate solutions from PMLE and MLE calculated by Multi-step EM and the EM algorithm.

are plotted as pairs in Figure 5.4. Most points fall in the bottom half of the plot, i.e., the MSE's of the MLE are larger than those of the PMLE. Therefore, PMLE gives more accurate parameter estimates. MSE's of both PMLE and MLE calculated via Multi-step EM are plotted in Figure 5.5. Most MSE's are on the 45−degree line, which indicates that Multi-step EM has significantly improved estimation accuracy for the MLE. The number of points above the diagonal are roughly equal to those below, so there is no large difference in the parameter accuracy between the PMLE and the MLE when Multi-step EM is used to do the parameter estimation.

The above discussion also tells us that even though Multi-step EM for MLE can converge to degenerate solutions some time, this algorithm has improved the accuracy of parameter estimation compared to the EM algorithm. Unlike the PMLE that requires careful choice of tuning parameters, the Multi-step EM algorithm is easily implemented.

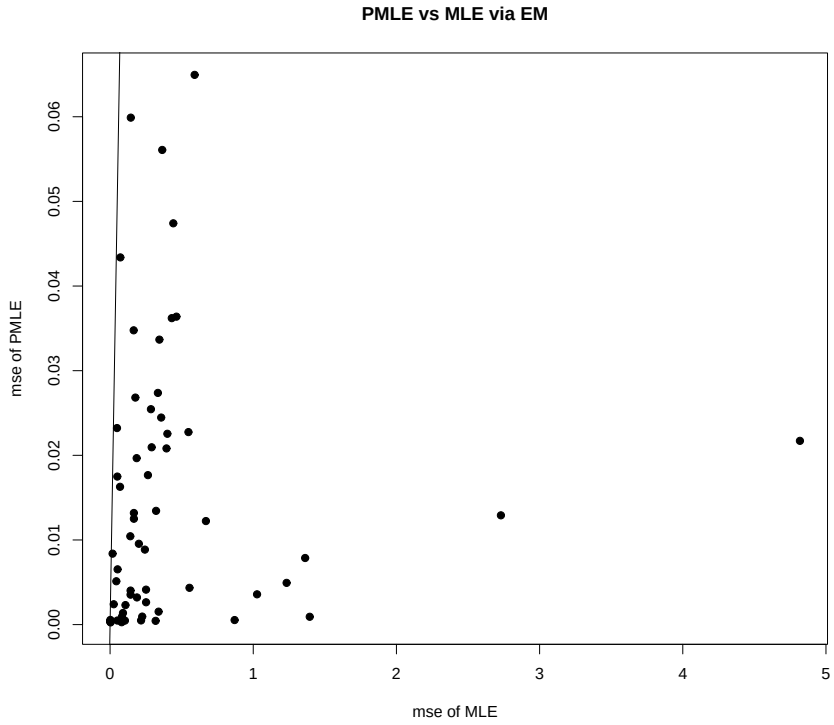

Figure 5.4: MSE of variances $(\sigma_{ij}^2, i = 1, 2, j = 1, 2)$ from PMLE and MLE calculated by EM for the first 16 simulation models. Degenerate solutions are included. There are in total 64 points in the plot. Each point represents a pair of MSE from PMLE and MLE of one variance parameter estimated via the EM algorithm.

In the following examples, the PMLE is estimated by the EM algorithm.

Consistency Testing

Since the number of components in the CMDA1 models is fixed, it is meaningful to investigate the bias and variance properties of individual parts of the PMLE. In this section, we generate data from two simulation models in Table 4.4 to illustrate the consistency property of the PMLE. Example 1 is a two-dimensional example sim-

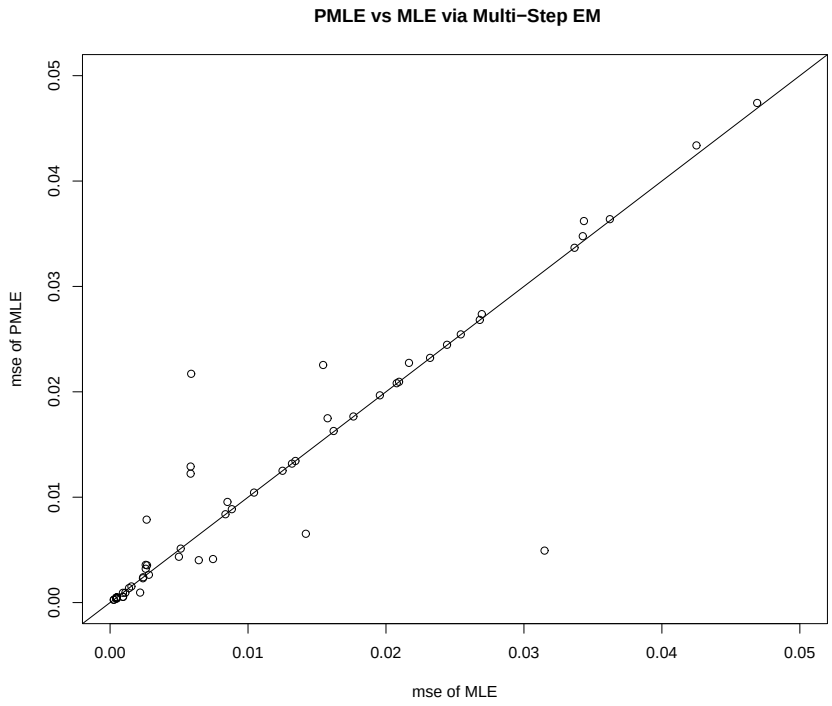

Figure 5.5: MSE of variances from PMLE and MLE calculated by Multi-step EM for the first 16 simulation models. Degenerate solutions are included.

ulated from the 12^{th} combination in Table 4.4, and Example 2 is a 10-dimensional example simulated from the 28^{th} combination. Both examples are among the more difficult classification problems in their own dimensionality. For Example 1, two sample sizes, $n = 70$ and $n = 280$ are considered. Similar to Example 1, two different sample sizes are also used to examine the consistency for Example 2: $n = 390$ and $n = 1,560$.

Example 1: A two-dimensional CMDA1 model with $\mathbf{\Psi} = (\pi_{11}^0, \pi_{21}^0, \mu_{11}^0, \mu_{12}^0, \mu_{21}^0, \mu_{22}^0, \mu_1^0, \mu_2^0, \sigma_{11}^0, \sigma_{12}^0, \sigma_{21}^0, \sigma_{22}^0, \sigma_1^0, \sigma_2^0) = (0.5, 0.5, 1.432, 0.501, -1.705, -1.463, -0.900, 1.533, 0.164, 0.379, 0.171, 1.036, 0.775, 0.102)$. The mean biases and standard deviations

Size	μ_{11}	μ_{12}	μ_{21}	μ_{22}	μ_1	μ_2
70	-0.091*	0.027	0.001	-0.008	0.008	-0.008*
	(0.254)	(0.235)	(0.030)	(0.189)	(0.146)	(0.027)
280	0.001	-0.007	0.000	0.003	0.001	-0.001
	(0.079)	(0.107)	(0.016)	(0.100)	(0.070)	(0.009)
	σ_{11}	σ_{12}	σ_{21}	σ_{22}	σ_1	σ_2
70	0.078*	-0.180*	-0.004*	-0.033*	-0.023*	0.020*
	(0.260)	(0.138)	(0.021)	(0.140)	(0.092)	(0.060)
280	-0.009	-0.023*	-0.002*	-0.002	0.000	0.000
	(0.068)	(0.077)	(0.011)	(0.067)	(0.053)	(0.016)
	π_{11}	π_{12}	π_{21}	π_{22}		
70	0.049*	-0.049*	0.001*	-0.001*		
	(0.112)	(0.112)	(0.006)	(0.006)		
280	0.003	-0.003	0.001*	-0.001*		
	(0.033)	(0.033)	(0.002)	(0.002)		

Table 5.3: Biases and standard deviations (in brackets) of parameter estimates for Example 1 using the PMLE with $P_1(\mathbf{\Psi})$. The biases that are significantly different from zero at a 5% significance level are indicated by $*$.

(in brackets) of parameter estimates are calculated from the 200 data sets simulated from the model and presented in Table 5.3.

The biases in Table 5.3 are computed in terms of $\mu_{11}-\mu_{11}^0$. When the sample size increases, all the biases and standard deviations decrease indicating the consistency of the PMLE for the CMDA1 model.

We are also interested in knowing which of these biases are significantly different from zero, i.e. whether the average estimates of the parameters are significantly different from the true parameters. With a large sample size (200 replicates) we

simply use a t-test. Table 5.3 indicates that all biases for the estimates of σ's and π's are significantly different from the true σ's and π's when the sample size is small. As the sample size increases, fewer parameter estimates are significantly different from the true parameters.

Example 2: A 10-dimensional CMDA1 model (the 28^{th} combination in Table 4.4). The parameters of true mixing distribution are listed in Table 5.4. All the π's are equal to 0.1 in this case. Due to the burden of computation, only 100 data sets are generated from the model. We list the biases and standard deviations of the PMLE of σ_{jk}'s and σ_j's in Table 5.5. As in the previous example, the bias and variance of the parameter estimates decrease as the sample size increases. A similar pattern may be seen in the location parameters and mixing proportions (not shown here).

Although we did not prove the asymptotic consistency for higher-dimensional CMDA1 models, this example suggests that the higher-dimensional CMDA1 models can be asymptotically consistent.

We also conduct the same hypothesis tests as in Example 1 to see if the biases of the parameter estimates are significantly different from the true parameters. As the sample increases, there are less biases that are significantly different from the truth (28 when the sample size is 390 vs. 23 when the sample size is 1560).

5.4.2 A Bayesian Perspective

In this section, we focus on using proper inverse gamma priors for the local variances as the penalty function. The prior for σ_{jk}^2 is an inverse gamma distribution, i.e. $\sigma_{jk}^2 \sim IG(\nu/2, \nu\lambda/2)$, and this is equivalent to specifying $\nu\lambda/\sigma_{jk}^2 \sim \chi_\nu^2$ with

μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}
0.405	-2.702	0.370	2.797	0.060
μ_{16}	μ_{17}	μ_{18}	μ_{19}	μ_{110}
1.206	-2.878	1.701	0.506	-1.096
μ_{21}	μ_{22}	μ_{23}	μ_{24}	μ_{25}
2.977	0.186	-2.341	0.799	1.788
μ_{26}	μ_{27}	μ_{28}	μ_{29}	μ_{210}
1.278	1.217	0.377	-1.447	0.042
μ_1	μ_2	μ_3	μ_4	μ_5
1.361	1.092	-2.777	0.109	0.787
μ_6	μ_7	μ_8	μ_9	μ_{10}
1.112	1.276	0.293	2.334	-0.098
σ_{11}	σ_{12}	σ_{13}	σ_{14}	σ_{15}
0.164	0.379	0.057	0.345	0.258
σ_{16}	σ_{17}	σ_{18}	σ_{19}	σ_{110}
0.034	0.369	0.292	0.108	0.128
σ_{21}	σ_{22}	σ_{23}	σ_{24}	σ_{25}
0.525	1.133	0.476	1.299	1.146
σ_{26}	σ_{27}	σ_{28}	σ_{29}	σ_{210}
0.109	0.739	0.953	0.127	0.876
σ_1	σ_2	σ_3	σ_4	σ_5
0.244	1.207	0.434	0.108	0.822
σ_6	σ_7	σ_8	σ_9	σ_{10}
1.015	0.678	1.236	0.136	0.183

Table 5.4: The true mixing distribution (μ's and σ's) for Example 2.

Sample	σ_{11}	σ_{12}	σ_{13}	σ_{14}	σ_{15}
390	-0.027	-0.007	-0.018*	-0.105*	-0.090*
	(0.169)	(0.358)	(0.021)	(0.121)	(0.213)
1560	0.088*	-0.005	-0.001	-0.016*	-0.037
	(0.197)	(0.260)	(0.010)	(0.071)	(0.256)
	σ_{16}	σ_{17}	σ_{18}	σ_{19}	σ_{110}
390	0.089*	-0.069*	-0.035*	-0.056*	-0.106*
	(0.208)	(0.202)	(0.306)	(0.051)	(0.048)
1560	0.051*	0.005	0.021*	-0.029*	-0.111*
	(0.199)	(0.187)	(0.050)	(0.044)	(0.036)
	σ_{21}	σ_{22}	σ_{23}	σ_{24}	σ_{25}
390	-0.030*	-0.655*	-0.251*	-0.422*	-0.616*
	(0.077)	(0.417)	(0.194)	(0.609)	(0.454)
1560	-0.014*	-0.258*	-0.153*	-0.333*	-0.300*
	(0.041)	(0.473)	(0.193)	(0.546)	(0.395)
	σ_{26}	σ_{27}	σ_{28}	σ_{29}	σ_{210}
390	0.197*	-0.499*	-0.499*	-0.005*	-0.179*
	(0.388)	(0.264)	(0.405)	(0.017)	(0.408)
1560	0.044*	-0.428*	-0.355*	0.000	-0.033
	(0.194)	(0.280)	(0.437)	(0.007)	(0.281)
	σ_1	σ_2	σ_3	σ_4	σ_5
390	0.008*	0.015*	0.008*	0.222*	0.028*
	(0.013)	(0.058)	(0.026)	(0.302)	(0.054)
1560	0.005*	0.011*	0.004*	0.158*	0.0176*
	(0.009)	(0.043)	(0.013)	(0.256)	(0.039)
	σ_6	σ_7	σ_8	σ_9	σ_{10}
390	-0.034*	-0.009*	-0.023*	0.034*	0.064*
	(0.056)	(0.030)	(0.058)	(0.041)	(0.067)
1560	-0.012*	0.001	-0.022*	0.025*	0.049*
	(0.029)	(0.016)	(0.031)	(0.042)	(0.047)

Table 5.5: Biases and standard deviations of the standard deviation estimates for Example 2 using the PMLE with $P_1(\mathbf{\Psi})$. The biases that are significantly different from zero at a 5% significance level are indicated by $*$.

$E(\sigma_{jk}^2) = \frac{\nu\lambda}{\nu-2}$ when $\nu > 2$ and $Var(\sigma_{jk}^2) = \frac{2\nu^2\lambda^2}{(\nu-2)^2(\nu-4)}$ for $\nu > 4$. This prior distribution is identical to the likelihood for σ_{jk}^2 arising from a data set with ν observations and sample variance λ. Then, the penalty function is

$$P_2(\mathbf{\Psi}) = \sum_{k=1}^{K}\sum_{j=1}^{P}\{-(\frac{\nu_{jk}}{2}+1)\log\sigma_{jk}^2 - \frac{\nu_{jk}\lambda_{jk}}{2\sigma_{jk}^2}\}, \tag{5.63}$$

The penalized complete data log-likelihood is

$$\begin{aligned} l_{pc}(\mathbf{\Psi}) = \sum_{k=1}^{K}\sum_{j=1}^{P}\Bigg\{\sum_{i\in C_k} z_{ijk}\Bigg[\log\pi_{jk} - \log\sigma_{jk} - \frac{(x_{ij}-\mu_{jk})^2}{2\sigma_{jk}^2} + \\ \sum_{l\neq j}(-\log\sigma_l - \frac{(x_{il}-\mu_l)^2}{2\sigma_l^2})\Bigg] - (\frac{\nu_{jk}}{2}+1)\log\sigma_{jk}^2 - \frac{\nu_{jk}\lambda_{jk}}{2\sigma_{jk}^2}\Bigg\}. \end{aligned} \tag{5.64}$$

Then the close form expressions for parameter estimates in the EM framework at the $(m+1)^{th}$ iteration are:

E-Step:

$$\begin{aligned} \hat{z}_{ijk}^{(m+1)} &= \hat{p}(\text{the } i^{th} \text{ observation} \in \text{the } j^{th} \text{ component}|\text{the } k^{th} \text{ class}) \\ &= \frac{\hat{\pi}_{jk}^{(m)} N(x_{ij};\hat{\mu}_{jk}^{(m)},\hat{\sigma}_{jk}^{(m)})\prod_{l\neq j} N(x_{il};\hat{\mu}_l^{(m)},\hat{\sigma}_l^{(m)})}{\sum_{m=1}^{P}\hat{\pi}_{mk}^{(m)} N(x_{im};\hat{\mu}_{mk}^{(m)},\hat{\sigma}_{mk}^{(m)})\prod_{l\neq m} N(x_{il};\hat{\mu}_l^{(m)},\hat{\sigma}_l^{(m)})} \end{aligned} \tag{5.65}$$

M-Step:

$$\hat{\pi}_{jk}^{(m+1)} = \frac{\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}}{\sum_{j=1}^{p}\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}}, \tag{5.66}$$

$$\hat{\mu}_{jk}^{(m+1)} = \frac{\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)} x_{ij}}{\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}}, \tag{5.67}$$

$$\hat{\sigma}_{jk}^{2\,(m+1)} = \frac{\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}(x_{ij} - \hat{\mu}_{jk}^{(m)})^2 + \nu_{jk}\lambda_{jk}}{\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)} + (\nu_{jk}+2)}, \tag{5.68}$$

$$\hat{\mu}_{l}^{(m+1)} = \frac{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)} x_{il}}{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}}, \tag{5.69}$$

$$\hat{\sigma}_{l}^{2(m+1)} = \frac{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}(x_{il} - \hat{\mu}_{l}^{(m)})^2}{\sum_{k=1}^{K}\sum_{j=1\&j\neq l}^{P}\sum_{i\in C_k} \hat{z}_{ijk}^{(m+1)}}. \tag{5.70}$$

The fact that $E(\sigma_{jk}^2) = \frac{\lambda_{jk}\nu_{jk}}{\nu_{jk}-2}$ for $\nu_{jk} > 2$ suggests that λ_{jk} should be chosen near the expected class specific variance σ_{jk}^2. In the absence of expert knowledge, some fraction of the sample variance along j^{th} direction of the k^{th} class could be used to choose λ_{jk}. We propose that

$$\lambda_{jk} = s_{jk}^2/25 = [\frac{\sum_{i\in C_k}(x_{ij} - \bar{x}_j)^2}{n_k - 1}]/25. \tag{5.71}$$

This represents the prior belief that standard deviation of the class specific variance will be roughly 1/5 of the sample standard deviation along j^{th} direction. Chipman (2006) used a similar idea when specifying a prior on the residual variance σ^2 in linear regression.

We choose $\nu_{jk} = 5$, for $j = 1, \ldots, P$ and $k = 1, \ldots, K$. This represents our belief that the prior distribution does not have a long tail and is centered around λ_{jk}. for $\nu_{jk} = 5$, the prior $(0.1, 0.5, 0.9)$ quantiles for σ_{jk}^2 are 0.54, 1.15 and 3.10 times λ_{jk} respectively.

For the penalty P_2, the same experiments as in the penalty P_1 are conducted to verify that the PMLE from P_2 does not converge to any degenerate solution. The comparisons between PMLE/Multi-step, MLE/Multi-step, PMLE/EM and MLE/EM gives the same conclusion as in P_1, i.e. there is no difference between PMLE/Multi-step and PMLE/EM; the PMLE gives more accurate parameter estimates than the MLE.

For the consistency testing of the PMLE under the penalty function P_2, the same Example 1 (combination 12) is used in the simulation. The simulation results (the biases and standard deviations of the parameter estimates of the PMLE) are presented in Table 5.6. The fact that all biases and standard deviations have gotten smaller with the larger samples suggests that MSE drops with increasing sample size and that PMLE's should be asymptotically consistent. Therefore, the PMLE of the CMDA1 model under the penalty P_2 is also asymptotically consistent. We also considered Example 2 for this penalty function. Results were similar to Section 5.4.1, and are not reproduced here.

5.5 Drug Discovery Data

We use the EM algorithm to estimate the PMLE of the CMDA1 model for the real drug data: NCI Antiviral AIDS Data using the first penalty P_1. The same 4 splits of the random samples as in Chapter 4 are used here. Performance is assessed by the AHR on the test set. The two different results (MLE/Multi-step and PMLE/EM) are presented in Table 5.7. Over the four replications, a paired t-test concludes that there is no significant difference between MLE/Multi-step and PMLE/EM at

Size	μ_{11}	μ_{12}	μ_{21}	μ_{22}	μ_1	μ_2
70	-0.139*	0.058*	0.784*	2.935*	-0.722*	-2.619*
	(0.485)	(0.328)	(0.192)	(0.445)	(0.133)	(0.421)
280	-0.024*	0.000	0.004*	0.034*	0.046*	-0.194*
	(0.121)	(0.115)	(0.018)	(0.139)	(0.069)	(0.081)
	σ_{11}	σ_{12}	σ_{21}	σ_{22}	σ_1	σ_2
70	-0.068*	-0.319*	-0.072*	-1.002*	-0.416*	1.197*
	(0.020)	(0.017)	(0.018)	(0.015)	(0.181)	(0.199)
280	-0.006*	-0.314*	-0.057*	-0.947*	-0.007	0.702*
	(0.003)	(0.013)	(0.001)	(0.001)	(0.050)	(0.201)
	π_{11}	π_{12}	π_{21}	π_{22}		
70	0.0183*	-0.018*	0.013*	-0.013*		
	(0.067)	(0.067)	(0.019)	(0.019)		
280	0.013*	-0.013*	0.002*	-0.002*		
	(0.041)	(0.041)	(0.003)	(0.003)		

Table 5.6: Biases and standard deviations (in brackets) of parameter estimates for Example 1 using the PMLE with $P_2(\mathbf{\Psi})$. The biases that are significantly different from zero at a 5% significance level are indicated by $*$.

Split	MLE/Multi-step	PMLE/EM
1	11.87	10.02
2	13.09	10.89
3	11.34	11.97
4	10.22	11.26
Average	11.63	11.03

Table 5.7: NCI data: AHR (%) for MLE/Multi-step and PMLE/EM.

Class	π_{1k}	π_{2k}	π_{3k}	π_{4k}	π_{5k}	π_{6k}
Active	0.13	0.25	0.08	0.20	0. 16	0.18
Inactive	0.04	0.07	0.25	0.27	0. 19	0.18

Table 5.8: The penalized estimates of mixing proportions estimated from Split2.

a 5% significance level.

Some interpretations of the model parameters may be possible. For example, large mixing proportions may be interpreted as evidence of the importance of the associated predictors. Consider the mixing proportions estimated from Split 2 shown in Table 5.8. It is interesting to see that both the active and inactive classes identify BCUT4 and BCUT6 as important variables, which are the same as the finding from Wang (2005).

Since both Multi-step EM and EM algorithms only give point estimates, it is difficult to do inference on the estimates. In future research, we can use some techniques to calculate standard errors for the estimates, which will be discussed in Chapter 6.

5.6 PMLE for the Second Order Model (CMDA2)

In this section, we consider a penalized maximum likelihood estimator for the CMDA2 model. As in Section 5.3.3, the penalized likelihood function is written as

$$Pl_n(\mathbf{\Psi}) = l_n(\mathbf{\Psi}) + p_n(\mathbf{\Psi}). \tag{5.72}$$

Before discussing the form of penalty term p_n, we review the form of the CMDA2 model, originally described in Section 4.10.

5.6.1 The CMDA2 Model

This model explores the two-dimensional subsets of descriptors. For P descriptors, there are $P(P-1)/2$ components in each class, as each component is specified by a pair of descriptors. The second order model can be written as

$$f(\mathbf{x}; \boldsymbol{\Psi_k}, \boldsymbol{\Psi_G}|y=k) = \sum_{j=1}^{P(P-1)/2} \pi_{jk} MVN(\mathbf{x_j}; \boldsymbol{\mu_{jk}}, \boldsymbol{\Sigma_{jk}}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l), \tag{5.73}$$

where $\boldsymbol{\Psi_k} = \{\mu_{\mathbf{jk}}, \boldsymbol{\Sigma_{jk}}\}_{j=1}^{P(P-1)/2}$ is the local parameter for class k and $\boldsymbol{\Psi_G} = \{\mu_l, \sigma_l\}_{l=1}^{P}$ is the global parameter. Here j indexes a pair of descriptors, i.e. $j = 1, \ldots, P(P-1)/2$ corresponds to pairs $\{(1,2),(1,3),\ldots,(P-1,P)\}$.

5.6.2 The Penalty Function for the CMDA2 Model

According to the multivariate analogues of Chen & Tan (2007), the penalty function for the CMDA2 model should have the following properties in order that the parameter estimates are asymptotically consistent:

C1. $p_n(\mathbf{\Psi}) = \sum_{k=1}^{K} \sum_{j=1}^{P(P-1)} \tilde{p}_n(\mathbf{\Sigma_{jk}})$.

C2. At any fixed $\mathbf{\Psi}$ such that $|\mathbf{\Sigma_{jk}}| > 0$ for all $j = 1, \ldots, P$ and $k = 1, \ldots, K$, we have $p_n(\mathbf{\Psi}) = o(n)$, and $\sup_{\mathbf{\Psi}} \max\{0, p_n(\mathbf{\Psi})\} = o(n)$. $p_n(\mathbf{\Psi})$ is differentiable with respect to $\mathbf{\Psi}$ and as $n \to \infty$, $p_n^{'}(\mathbf{\Psi}) = o(\sqrt{n})$ at any fixed $\mathbf{\Psi}$ such that $|\mathbf{\Sigma_{jk}}| > 0$ for all $j = 1, \ldots, P$ and $k = 1, \ldots, K$.

C3. For large enough n, $\tilde{p}_n(\mathbf{\Sigma}) \leq 4(\log n)^2 \log |\mathbf{\Sigma}|$, when $|\mathbf{\Sigma}|$ is smaller than cn^{-2} for some $c > 0$.

An additive function is used as the penalty function, so C1 simplifies the numerical computation. C2 limits the effect of penalty. C3 means that the penalty will counteract both $\sigma_j = 0$ and $\mathbf{\Sigma}$ that are degenerate such as a $\mathbf{\Sigma}$ corresponding to variables with correlations of ± 1. This is a new kind of degeneracy not encountered in the CMDA1 model.

5.6.3 PMLE for the NCI Antiviral AIDS Data

In the section, the penalized CMDA2 model is applied on the NCI Antiviral Aids data to obtain the penalized maximum likelihood estimates. We use the Wishart distribution to generate a penalty function for $\mathbf{\Sigma_{jk}}$ taking $p_n(\mathbf{\Psi})$ as the log of a $W_2(\boldsymbol{S_{jk}}, n_k) = \frac{|\mathbf{\Sigma_{jk}}|^{(n_k-3)/2}}{2^{n_k}|\boldsymbol{S_{jk}}|^{n_k/2}\Gamma_2(n_k/2)} \exp(-\frac{1}{2} tr(\boldsymbol{S_{jk}}^{-1}\mathbf{\Sigma_{jk}}))$. That is,

$$p_n(\mathbf{\Psi}) = -\sum_{k=1}^{K} D_k \sum_{j=1}^{P(P-1)/2} \{tr(\boldsymbol{S_{jk}}^{-1}\mathbf{\Sigma_{jk}}) + \log |\mathbf{\Sigma_{jk}}|\}, \tag{5.74}$$

where $\boldsymbol{S_{jk}}$ is the mode of the prior distribution. $D_k \in \Re^+$ is a tuning parameter, whose increasing values implies a stronger concentration of the density near $\boldsymbol{S_{jk}}$.

The penalized maximum likelihood estimates in EM framework are:

E-Step:

$$\hat{z}_{ijk} = \frac{\hat{\pi}_{jk} MVN(\mathbf{x_j}; \mu_{\mathbf{jk}}, \boldsymbol{\Sigma_{jk}}) \prod_{l \neq j} N(x_l; \mu_l, \sigma_l)}{\sum_{j^*=1}^{p(p-1)/2} \hat{\pi}_{j^*k} MVN(\mathbf{x_{j^*}}; \boldsymbol{\mu_{j^*k}}, \boldsymbol{\Sigma_{j^*k}}) \prod_{l \neq j^*} N(x_l; \mu_l, \sigma_l)}; \tag{5.75}$$

M-Step:

$$\hat{\pi}_{jk} = \frac{\sum_{i \in C_k} \hat{z}_{ijk}}{\sum_{l=1}^{P(P-1)/2} \sum_{i \in C_k} \hat{z}_{ilk}}; \tag{5.76}$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{jk}} = \frac{\sum_{i \in C_k} \hat{z}_{ijk} \mathbf{x_{ij}}}{\sum_{i \in C_k} \hat{z}_{ijk}}; \tag{5.77}$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{jk}} = \frac{\sum_{i \in C_k} \hat{z}_{ijk} (\mathbf{x_{ij}} - \hat{\boldsymbol{\mu}}_{\boldsymbol{jk}})(\mathbf{x_{ij}} - \hat{\boldsymbol{\mu}}_{\boldsymbol{jk}})^T + 2D_k \boldsymbol{S_{jk}}}{\sum_{i \in C_k} \hat{z}_{ijk} + 2D_k}; \tag{5.78}$$

$$\hat{\mu}_l = \frac{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk} x_{il}}{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk}}; \tag{5.79}$$

$$\hat{\sigma}_l^2 = \frac{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk} (x_{il} - \hat{\mu}_l)^2}{\sum_{k=1}^{K} \sum_{i \in C_k} \sum_{j \neq l}^{P(P-1)/2} \hat{z}_{ijk}}. \tag{5.80}$$

In the application, we assume $D_k = D$, $k = 1, \ldots, K$. A sequence of D, i.e. $D = (1.5, 2, 3, 4, 5, 6, 7, 8)$, is used to identify the best one, which gives highest AHR for the testing set. We find that D in the range of $(1.5, 8)$ returns similar values of AHR, so $D = 1.5$ is chosen in the computation. We choose $\boldsymbol{S_{jk}}$ to be the sample covariance matrix of the two class-specific descriptors of the j^{th} component. That is, training data with $y = k$ are used for the sample covariance of x_{j_1} and x_{j_2}, when $j = (j_1, j_2)$.

As before, the CMDA2 model is applied on the NCI antiviral AIDS Data with four splits. The data are transformed using normal distribution transformation, i.e. $\frac{1}{s}\phi(\frac{x-\bar{x}}{s})$, where $\bar{x}$ is the sample mean and s is the sample standard deviation.

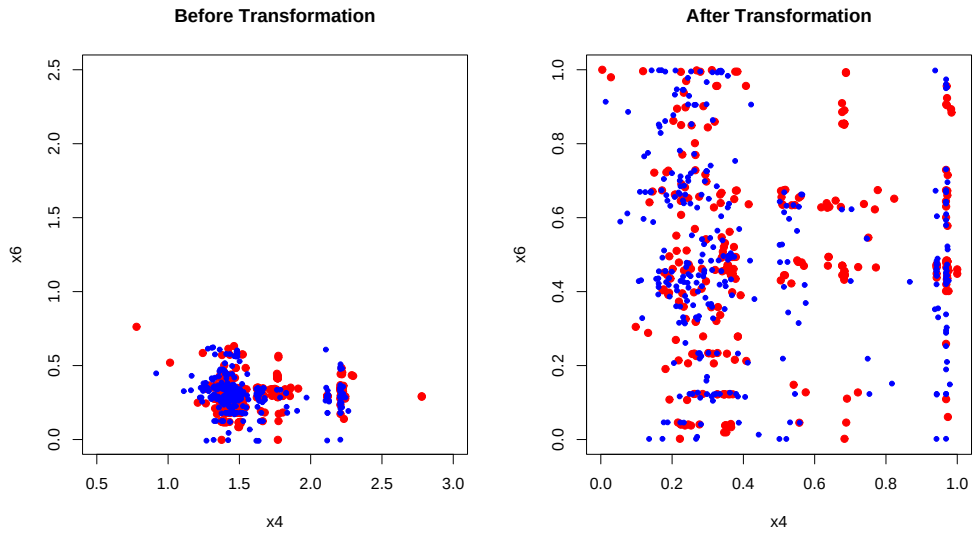

Figure 5.6: Normal transformation of the subspace (x_4, x_6) of the NCI data with 300 active and 300 inactive compounds: Left, before transformation; Right, after transformation. Red and blue represent two classes.

Figure 5.6 shows the data before and after the transformation. It is clear that after transformation, outliers are pushed toward the centre of the data, so the effect of outliers can be reduced through transformation.

We apply both the CMDA2 model and the MclustDA model with 15 components to the NCI Antiviral AIDS Data with the 4 same training-testing splits as in previous sections. The training and test sets each have $n = 14,906$ compounds, of which 304 are active compounds. We conduct four experiments (4 splits), which will be referred to as "Split 1", ..., "Split 4" in the text below. Performance is assessed by the AHR on the test set. The AHR's returned from both approaches for the 4 splits are listed in Table 5.9. A paired t-test is conducted and the $p-value = 0.521$ indicates that there is no statistically significant difference between the penalized

CMDA2 and MclustDA.

Compared to the AHR's returned by the CMDA1 model, the CMDA2 model has higher AHR's for the four different splits. Therefore, the CMDA2 model does have a better performance than the CMDA1 model. A paired t-test is also performed and the $p-value = 0.036$ confirms the previous conclusion that the CMDA2 model outperforms the CMDA1 model.

Furthermore, the CMDA2 model may give insight into the data. For example, mixture components with large weights represent large proportions of the data. Since each component is associated with a two-dimensional subspace, a large weight π implies that data are concentrated in this subspace, and that the subspace may be useful for discriminating active compounds. Table 5.10 lists the estimates of the mixing proportions for the second split of the NCI data. The active class identifies the descriptor pairs $(1, 4)$, $(2, 4)$ and $(3, 4)$ as important mechanisms, and the inactive class identifies $(1, 2)$, $(2, 4)$ and $(3, 4)$ as important mechanisms.

It is interesting to note the prevalent role that x_4 plays in the active components, as this was identified previously by Wang (2005) as the most important predictor.

5.7 Discussion

In this chapter, the PMLE for the two-dimensional CMDA1 model has been proven to be asymptotically consistent, which is confirmed by the simulations. Although the consistency for higher-dimensional CMDA1 models has not been proved, the simulation results suggest that high-dimensional CMDA1 models may also be asymp-

Split	CMDA1	CMDA2	MclustDA(15)
1	11.87	17.76	22.14
2	13.09	14.64	16.27
3	11.34	16.91	13.49
4	10.22	19.20	21.39
Average	11.63	17.13	18.32

Table 5.9: AHR (%) for the CMDA1 model, the PMLE of the CMDA2 model and the MclustDA model with 15 components.

Subspaces	(x_1, x_2)	(x_1, x_3)	(x_1, x_4)	(x_1, x_5)	(x_1, x_6)
Active	0.078	0.043	0.211	0.028	0.063
	(x_2, x_3)	(x_2, x_4)	(x_2, x_5)	(x_2, x_6)	(x_3, x_4)
	0.095	0.180	0.025	0.005	0.196
	(x_3, x_5)	(x_3, x_6)	(x_4, x_5)	(x_4, x_6)	(x_5, x_6)
	0.034	0.006	0.020	0.009	0.007
Subspaces	(x_1, x_2)	(x_1, x_3)	(x_1, x_4)	(x_1, x_5)	(x_1, x_6)
Inactive	0.177	0.029	0.093	0.043	0.068
	(x_2, x_3)	(x_2, x_4)	(x_2, x_5)	(x_2, x_6)	(x_3, x_4)
	0.077	0.151	0.049	0.038	0.160
	(x_3, x_5)	(x_3, x_6)	(x_4, x_5)	(x_4, x_6)	(x_5, x_6)
	0.052	0.012	0.016	0.011	0.024

Table 5.10: The estimates of mixing proportions corresponding to each subspace from Split2.

totically consistent.

For the NCI data, the CMDA2 model has better performance than the CMDA1 model by generating higher AHR. This indicates that the CMDA2 model can represent the data better than the CMDA1 model. However, we also find that the CMDA2 model is very sensitive to the initial values, i.e. different initial values give different parameter estimates although the AHR's obtained are very close. This may occur because the CMDA2 model can only catch one mechanism in each subspace (i.e. a single normal mixture component). This may not be sufficient to catch the whole structure of the NCI data.

Choosing reasonable values of the tuning parameters for the penalty functions becomes an important question in the PMLE approach. The variances or covariance matrices of the subspaces of the data and the assessment of sensitivity of results to different penalization parameter values can provide guidance to choose the right values for the tuning parameters.

The fact that the performance of the MclustDA model with 15 components is much better than that of the MclustDA model with 6 components indicates that adding extra components may improve the performance of the CMDA1 model. Furthermore, the CMDA2 model performed better than the CMDA1 model. This indicates that both the number of components and two-dimensional component structure of the CMDA2 model are helpful in ranking active compounds.

Chapter 6

Future Research

This chapter is divided into three parts. Section 6.1 describes the future research of Cluster Structure-Activity Relationship (CSARA) analysis. The future research for the CMDA model and the PMLE are presented in Section 6.2 and Section 6.3 respectively.

6.1 CSARA

In the analysis of the CSARA method, the large number of clusters (e.g. $3,000$, $5,000$, $10,000$, etc) were used in the study. Exactly one compound was randomly selected from each cluster to act as the training data regardless of cluster size. The response from a single compound will be quite variable. A more stable approach might be to reduce the number of clusters and select more compounds per cluster. Also the approach will allow the number of compounds sampled to vary according to cluster size. The modification will increase the chance of identifying more active

compounds as a large sample from the cluster should provide more information on cluster features. Therefore, the problem how to choose multiple compounds from each cluster arises.

It is also interesting to note that with multiple samples per cluster, some ranking of clusters may be possible. The current approach of assaying one compound per cluster yields only a 0/1 activity label, which makes it impossible to predict the probability of active compounds in each cluster and difficult to rank clusters according to the predicted activity probabilities. However, sampling multiple compounds can make it possible to estimate a proportion of actives and use this for cluster ranking.

6.2 CMDA

The challenges presented by QSAR modeling have been listed in Section 1.3: (1) unbalanced response or rare target problem, (2) multiple mechanisms, (3) subspace-governed activity, (4) nonlinear relationship among descriptors and (5) measurement errors. In Chapter 4, the primary focus was the CMDA1 model, which handles the first three challenges. For the fourth challenge of nonlinear relationship, the CMDA2 model may provide better representation of drug discovery data as it can represent bivariate dependencies within mixture component. A more careful and systematic simulation design is needed to explore the characteristics of the CMDA2 model. A challenge encountered with applying the current implementation of the CMDA2 model to drug discovery data is the larger impact of outliers not due to measurement errors. Modifications of the method capable of dealing with outliers

will be an important extension. The large number of mixture components and bivariate dependence structure of the CMDA2 model makes computations very intensive. Efficient algorithms using parallel computing need to be developed.

We also want to test the order of CMDA models, i.e. the number of components in each class. For example, the CMDA1 model can have more than P components. Identifying the number of components in mixture models is an important topic, which is related to model selection techniques. In the mixture context, such model selection can be implemented by seeking to set some mixing proportions to be zero or to constrain other parameters across mixtures. One possible approach is the use of Lasso-type penalties (Tibshirani 1996 and Fan & Li 2001). In the future research, we would like to include this feature into our current algorithm and do model selection automatically.

We notice that selecting or identifying important variables for a drug discovery data set is meaningful. This problem can be viewed as a kind of model selection. The techniques described in the previous paragraph may be relevant, since each component of both the CMDA1 and CMDA2 models is associated with one or two particular variables.

In the thesis, the CMDA model has been applied only to two-class data. The CMDA model may require slight changes in order to be applied to the data with multiple classes. Further, in order to incorporate various descriptors (e.g. categorical, ordinal, etc.), the component densities can be discrete, and other types of density functions.

Usually, drug discovery data can have hundreds descriptors. In such high di-

mensions, the CMDA model may not work as well due to the large number of parameters and the complexity of the model structure. So some sort of dimension reduction strategy may need to be used under this context. Some algorithms, such as recursive partitioning and random forest, can preliminarily determine the optimal subset of descriptors to reach the aim of dimension reduction.

6.3 Penalized Maximum Likelihood Estimation

Chapter 5 proves the asymptotic consistency of the PMLE for two-dimensional multivariate normal mixtures with diagonal covariance matrices. In future research, we want to prove the asymptotic consistency of the PMLE for higher dimensional multivariate normal mixtures without independence constrains. The tentative approach may be calculating the largest eigenvalue for each component given the class label k, then counting the number of observations dropping in a small neighbourhood of the local parameter along the dimension of the largest eigenvalue. The proof should be similar to the approach taken in Chapter 5.

For both the PMLE and the MLE, the EM algorithm only gives point estimates without any uncertainty measurements, so it is not currently possible to make inference for the estimates from two approaches. Later, we will modify the multi-step EM algorithm by adding one more step, i.e. calculating the covariance matrix for the MLE or the PMLE. Bootstrap techniques can be also used to get empirical uncertainty for parameter estimates. Finally, we also can try a Markov Chain Monte Carlo (MCMC) approach to quantify parameter uncertainty (Bensmail, Celeux, Raftery & Robert (1997) and Richardson & Green (1997)). Since PMLE's can be

viewed as the estimates using prior distributions, a Bayesian approach to inference would be a natural extension.

Appendix A: Related Theories

Theorem 6 (Dominated Convergence Theorem) *Lebesgue's dominated convergence theorem states that if a sequence* $\{f_n : n = 1, 2, \ldots\}$ *of real-valued measurable functions on a measurable space* S *converges almost everywhere, and is "dominated" by some nonnegative function* g *in* L'*, then*

$$\int_S \lim_{n\to\infty} f_n = \lim_{n\to\infty} \int_S f_n \tag{6.1}$$

i.e. $|f_n(x)| \leq g(x)$ *for every* n *and almost every* x *(i.e. the measure of the set of exceptional values of* x *is zero).* g *in* $L\prime$ *means* $\int_S |g(x)| < \infty$

Theorem 7 (Kolmogorov's Strong Law of Large Numbers) *Let* $\{X_i\}$ *be I.I.D. The existence of a finite constant* c *for which*

$$\frac{1}{n}\sum_{i=1}^{n} X_i \rightarrow^{wp1} c \tag{6.2}$$

holds if and only if $E\{X_i\}$ *is finite and equals* c*.*

Theorem 8 (Bernstein's Inequality) *Let* $Y_1, \ldots, Y_n$ *be independent random variables satisfying* $P(|Y_i - E\{Y_i\}| \leq m) = 1$*, each* i*, where* $m < \infty$*. Then for* $t > 0$

$$P(|\sum_{i=1}^{n} Y_i - \sum_{i=1}^{n} E\{Y_i\}| \geq nt) \leq 2\exp(-\frac{n^2t^2}{2\sum_{i=1}^{n} var\{Y_i\} + \frac{2}{3}mnt}) \tag{6.3}$$

for all $n = 1, 2, \ldots$.

Theorem 9 (Borel-Cantelli Lemma) **(i)** *For arbitrary events* B_n, *if* $\sum_n P(B_n) < \infty$, *then* $P(B_n$ *infinitely often*$) = 0$.

(ii) *For independent events* B_n, *if* $\sum_n P(B_n) = \infty$, *then* $P(B_n$ *infinitely often*$) = 1$.

Bibliography

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

Bensmail, H. & Celeux, G. (1996), 'Regularized gaussian discriminant analysis through eigenvalue decomposition', *Journal of the American Statistical Association* **91**, 1743–1948.

Bensmail, H., Celeux, G., Raftery, A. E. & Robert, C. P. (1997), 'Inference in model-based cluster analysis', *Statistics and Computing* **7**, 1–10.

Blischke, W. R. (1962), 'Moment estimators for the parameters of a mixture of two Binomial distributions', *Annals of Mathematical Statistics* **33**, 444–454.

Blischke, W. R. (1964), 'Estimating the parameters of mixtures of Binomial distributions', *Journal of the American Statistical Association* **59**, 510–528.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and Regression Trees*, Wadsworth Publishing Co. Inc.

Bremmer, J. M. (1978), 'Algorithm AS 123: Mixtures of Beta distributions', *Applied Statistics* **27**, 104–109.

Burden, F. (1989), 'Molecular identification number for substructure searches', *Journal of Chemical Information and Computer Sciences* **29**, 225–227.

Chen, J. & Tan, X. (2007), 'Inference for multivariate normal mixture model', *Manuscript*.

Chen, J., Tan, X. & Zhang, R. (2007), 'Inference for normal mixtures in mean and variance', *Statistica Sinica*.

Chipman, H. (2006), Prior distributions for bayesian analysis of screening experiments, *in* A. Dean & S. Lewis, eds, 'Screening: Methods for Experimentation in Industry, Drug Discovery, and Genetics', Springer.

Ciuperca, G., Ridolfi, A. & Idier, J. (2003), 'Penalized maximun likelihood estimator for normal mixtures', *Scandinavian Journal of Statistics* **30**, 45–59.

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

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

Drews, J. (2000), 'Drug discovery: A historical perspective', *Science* **287**, 1960–1965.

Dunbar, J. B. (1997), 'Cluster-based selection', *Perspectives in Drug Discovery and Design* **7/8**, 51–63.

Engels, M. F. M. & Venkatarangan, P. (2001), 'Smart screening: Approaches to efficient HTS', *Current Opinion in Drug Discovery and Development* **4**, 275–283.

Fan, J. & Li, R. (2001), 'Variable selection via nonconcave penalized likelihood and its oracle properties', *Journal of the American Statistical Association* **96**, 1348–1360.

Feng, J., Lurati, L., Ouyang, H., Robinson, T., Wang, Y. Y., Yuan, S. & Young, S. S. (2003), 'Predictive toxicology: Benchmarking molecular descriptors and statistical methods', *Journal of Chemical Information and Computer Sciences* **43**, 1463 –1470.

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

Friedman, J. (1989), 'Regularized discriminant analysis', *Journal of the American Statistical Association* **84**, 165–175.

Green, P. J. (1990), 'Bayesian reconstructions from emission tomography data using a modified EM algorithm', *IEEE Transactions on Medical Imaging* **9**, 84–93.

Hartigan, J. A. & Wong, M. A. (1979), 'Algorithm AS 136 A K-means clustering algorithm', *Applied Statistics* **28**, 100–108.

Hastie, T. & Tibshirani, R. (1996), 'Discriminant Analysis by Gaussian Mixtures', *Journal of the Royal Statistical Society* **58**, 155–176.

Hathaway, R. J. (1985), 'A constrained formulation of maximum likelihood estimation for normal mixture distributions', *Annals of Statistics* **13**, 795–800.

Hawkins, D. M. & Kass, G. V. (1982), *Automatic Interaction Detection in Topics in Applied Multivariate Analysis*, Cambridge University Press.

Kao, J. H. K. (1959), 'A graphical estimation of mixed Weibull parameters in life-testing electron tubes', *Technometrics* **1**, 389–407.

Kiefer, J. & Wolfowitz, J. (1956), 'Consistency on the maximum likelihood estimator in the presence of infinitly many incidental parameters', *The Annals of Mathematical Statistics* **27**, 887–906.

Lajiness, M. S. (1997), 'Dissimilarity-based compound selection techniques', *Perspectives in Drug Discovery and Design* **7**, 65–84.

Lam, R. L. H. (2001), *Design and Analysis of Large Chemical Databases for Drug Discovery*, Ph.D thesis, Department of Statistics and Actuarial Science, University of Waterloo, Canada.

Lam, R. L. H., Welch, W. J. & Young, S. S. (2002), 'Uniform coverage designs for molecule selection', *Technometrics* **44**, 99–109.

Lindsay, B. G. (1995), *Mixture Models: Theory, Geometry and Applications*, Vol. 5, NSF-CBMS Regional Conference Series in Probability and Statistics.

MacQueen, J. (1967), 'Some methods for classification and analysis of multivariate observations', *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press* **1**, 281–297.

Marriott, F. H. C. (1975), 'Separating mixtures of normal distributions', *Biometrics* **31**, 767–769.

McLachlan, G. J. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, Wiley.

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

McLachlan, G. J. & Krishnan, T. (1997), *The EM Algorithm and Extensions*, Wiley.

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

Pearlman, R. S. & Smith, K. M. (1999), 'Metric validation and the receptor-relevant subspace concept', *Journal of Chemical Information and Computer Sciences* **39**, 28–35.

Pearson, K. (1894), 'Contributions to the mathematical theory of evolution', *Philosophical Transactions* **A 185**, 71–110.

R Development Core Team (2006), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3.

Redner, R. A. (1981), 'Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions', *Annals of Statistics* **9**, 225–228.

Richardson, S. & Green, P. J. (1997), 'On bayesian analysis of mixtures with an unknown number of components', *Journal of the Royal Statistical Society* **B 59**, 731–792.

Rider, P. R. (1961), 'The method of moments applied to a mixture of two exponential distributions', *Annals of Mathematical Statistics* **32**, 143–147.

Rider, P. R. (1962), 'Estimating the parameters of mixed Poisson, Binomial, and Weibull distributions by the method of moments', *Bulletin of the International Statistical Institute* **39**, 225–232.

Ridolfi, A. & Idier, J. (1999), Penalized maximum likelihood estimation for univariate normal mixture distributions, In Actes du 17e colloque GRETSI, Vannes, France, pp. 259–262.

Ridolfi, A. & Idier, J. (2000), Penalized maximum likelihood estimation for univariate normal mixture distributions, Bayesian inference and maximum entropy methods, Maxent Workshops. Gif-sur-Yvette, France, Juli 2000.

Simon, J. A., Dunstan, H., Lamb, J. R., Evans, D. R., Cronk, M. & Irvine, W. (2000), 015 Yeast as a model organism for anticancer drug discovery: An update from the NCI/Fred Hutchinson Cancer Research Center collaboration, *in* 'Proceedings of the 11^{th} NCI · EORTC · AACR Symposium'.

Symons, M. J. (1981), 'Clustering criteria and multivariate normal mixtures', *Biometrics* **37**, 35–43.

Teicher, H. (1963), 'Identifiability of finite mixtures', *The Annals of Mathematical Statistics* **34**, 1265–1269.

Therneau, T. M. & Atkinson, B. (2006), *rpart: Recursive Partitioning.* R package version 3.1-29, R port by Brian Ripley.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **58**, 267–288.

Todeschini, R. & Consonni, V. (2000), *Handbook of Molecular Descriptors*, Wiley-VCH.

Valler, M. J. & Green, D. (2000), 'Diversity screening versus focussed screening in drug discovery', *Drug Discovery Today* **5**, 286–293.

Wang, Y. Y. (2005), *Statistical Methods for High Throughput Screening Drug Discovery Data*, Ph.D thesis, Department of Statistics and Actuarial Science, University of Waterloo, Canada.

Welch, W. J. (2002), *Computational Exploration of Data Course Notes*, Department of Statistics and Actuarial Science, University of Waterloo.

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

Yakowitz, S. & Spragins, J. (1968), 'On the identifiability of finite mixtures', *The Annals of Mathematical Statistics* **39**, 209–214.

Young, S. S. & Hawkins, D. M. (1998), 'Using recursive partitioning to analyze a large SAR data set', *Structure-Activity Relationship and Quantitative Structure-Activity Relationship* **8**, 183–193.

Young, S. S., Lam, R. L. H. & Welch, W. J. (2002), 'Initial compound selection for sequential screening', *Current Opinion in Drug Discovery and Development* **5**, 422–427.