Mixture Modeling
for
Multivariate Observations

Xuxu Wang

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
The Department of Statistics
The University of Auckland, New Zealand
2015
Completing the PhD and writing this thesis was an amazing journey and a life-changing experience for me. It would not have been possible to do without the support and guidance that I received from my supervisor, Dr. Yong Wang.

With the greatest pleasure, I would like to express my sincere appreciation and thanks to Dr. Yong Wang for his fundamental role in my doctoral work. In the first year of my PhD study, Yong provided me with every bit of guidance, assistance, and expertise that I needed to start to learn to be a researcher. From the second year, after I ventured into research, Yong gave me enormously valuable inspirations, advices, feedbacks and encouragements. Whenever I wanted a discussion about my research, the only thing I need to do was knocking on Yong’s door because I knew that he would help me without reserve. His patience, motivation, enthusiasm, and immense knowledge helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

A special thank you to my family. Words can not express how grateful I am to my dear grandma Qunfang Kang, dear mum Jinjian Lu and dear dad Jianbo Wang for all of the loves and supports that you gave me. Your understandings, advises and indescribable supports to me throughout my whole life are invaluable. I am very grateful about your care, your love and trust in me. All other family members and friends, thank you for your support!
I am very grateful to all those at the department, especially Associate Professor Brian McArdle, Associate Professor Ross Ihaka, Dr. Thomas Yee and others who were always so helpful and provided me with their assistance throughout my dissertation.
Abstract

In this thesis, nonparametric multivariate density estimation is studied. The kernel density estimation predominately used in the literature is known to be less than ideal in both computation and estimation efficiency, especially for multivariate observations. Mixtures, particularly nonparametric and semiparametric mixtures, have the potential to outperform the kernel-based methods, as has been shown in previous research in the univariate case. It is the main goal of this research to extend the application of these mixture models to estimate nonparametrically a multivariate distribution. The major difficulty with the likelihood approach that is associated with the estimation of these mixtures in the multivariate case is that the likelihood function can not be maximized directly with respect to the whole component covariance matrix, because the covariance matrix will become singular, and that it is also computationally infeasible to use a model selection method to select a large part of the covariance matrix. To overcome it, the new method uses a volume parameter $h$ to put a minimal restriction on the covariance matrix and hence has the likelihood function bounded with $h$ fixed. The scalar $h$ plays the same role as a bandwidth parameter in the univariate case and its value can be determined by a model selection method. New algorithms are also developed for fitting nonparametric and semiparametric mixtures in the multivariate case. Empirical studies using simulated and real-world data show that the new method performs significantly better than
the kernel-based methods.

The availability of the new density estimator makes it possible to provide new solutions to other statistical problems. Two problems are especially addressed in the thesis: classification and the fitting of generalized linear mixed models with multiple random effects. For classification, a new classification method is proposed that makes a direct use of the new density estimator. It firstly estimates the probability density function for the observations in each class and then classifies a new observation according to the maximum a posterior criterion. Some issues specific to classification are also studied and discussed. The new classification method is able to produce adaptively smooth and complicated decision boundaries in a high-dimensional space. Empirical studies indicate that it outperforms the commonly used methods. For generalized linear mixed models, the situation with multiple random effects is investigated, where their joint distribution is completely unspecified. This naturally leads to a semiparametric multivariate mixture formulation and its fitting is to find the nonparametric maximum likelihood estimate of the joint distribution of random effects and the maximum likelihood estimate of the fixed effects. Efficient algorithms are also developed for fitting these generalized linear mixed models. Numerical studies with simulation and real-world data show that the new method performs favorably well as compared to the parametric method under the normality assumption.
Contents

Acknowledgment i

Abstract iii

1 Introduction 1
   1.1 Nonparametric Density Estimation ......................... 1
   1.2 Using Mixtures for Density Estimation .................... 3
   1.3 Multivariate Density Estimation and Its Usage ............ 5
   1.4 Contributions ............................................ 9
   1.5 Outline .................................................. 10

2 Literature Review 13
   2.1 Kernel Density Estimation ................................. 13
   2.2 Mixture Models ........................................... 15
   2.3 Maximum Likelihood Estimation of Mixture Models .......... 17
   2.4 Mixture-based Density Estimation .......................... 20
   2.5 Multivariate Density Estimation ............................ 22
   2.6 Usage of Density Estimation ............................... 24
   2.7 Algorithms for Fitting Mixture Models ..................... 28
3 Nonparametric Multivariate Density Estimation Using Mixtures

3.1 Introduction ..................................................... 33

3.2 Density Estimation using Multivariate Nonparametric/Semiparametric mixtures ............................................. 34

3.2.1 Nonparametric formulation .................................. 34

3.2.2 Semiparametric formulation .................................. 36

3.2.3 Volume selection ................................................ 37

3.3 Computation ....................................................... 39

3.3.1 A hybrid approach ............................................. 39

3.3.2 Updating $\pi$, $\Theta$ and $B$ .................................... 39

3.3.3 Updating $G$ ...................................................... 41

3.3.4 The hybrid algorithm .......................................... 46

3.4 Demonstration of finding an NPMLE ................................ 47

3.5 Concluding Remarks .............................................. 48

4 Numerical Studies for Nonparametric Multivariate Density Estimation 51

4.1 Simulation Studies ................................................. 51

4.1.1 Setup .......................................................... 51

4.1.2 Normal mixtures .............................................. 54

4.1.3 Mixtures of $t$-distributions .................................. 57

4.2 Real Data Sets ..................................................... 59

4.2.1 Setup .......................................................... 59

4.2.2 The Indian Liver Patient data ................................ 61

4.2.3 The Wine data ................................................. 64

4.3 Concluding Remarks .............................................. 68

5 Classification Using Semiparametric Mixtures 71

5.1 Introduction ....................................................... 71

5.2 Classification Based on Density Estimation ...................... 73
6 Generalized Linear Mixed-Effects Models: A Semiparametric Mixture Solution

6.1 Introduction ................................................. 99
6.2 Generalized Linear Mixed Models .......................... 101
6.3 A Semiparametric Mixture .................................. 103
6.4 Computation .................................................. 104
   6.4.1 Updating $G$ ....................................... 105
   6.4.2 Updating $(G, \beta)$ ................................. 109
6.5 Demonstration of Algorithm 5 ............................. 111
6.6 The Direct Search Directional Derivative Method ....... 113
6.7 Difference between two methods for fitting semiparametric mixtures . 118
6.8 Summary .................................................... 121

7 Numerical Studies of Generalized Linear Mixed-Effects Models 123

7.1 Introduction .................................................. 123
7.2 Simulation Studies .......................................... 123
   7.2.1 Bivariate Random Effects .......................... 124
7.2.2 Multivariate Random Effects . . . . . . . . . . . . . . . . . . . 130
7.3 Real Data Set . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
7.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138

8 Summary and Future Works 139

8.1 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
8.2 Future Works . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 141
Chapter 1

Introduction

1.1 Nonparametric Density Estimation

Density estimation is a fundamental problem in statistics. It refers to the construction of an estimate of the density function, or broadly speaking the distribution function, from observations. As a natural tool to investigate and explore the hidden features of observations, density estimation enjoys many applications. For example, one can be interested in accessing the capability of a stable manufacturing process (Polansky, 2014), or in obtaining poverty estimates and describing the global income distribution (Minoiu and Reddy, 2014), or in forecasting the volatility for the Standard & Poor’s 500 daily returns series (Almeida et al., 2014). Solving these problems can all be directly related to density estimation. Successful density estimation can also assist solving other types of statistical problems, such as clustering and classification.

There are two major approaches to estimating a density function. One, of course, is parametric, where a limited number of parameters for the density function is assumed and the parameters of the function are then chosen optimally by fitting the model to observations. A simple example is to find the mean and variance of a normal distribution. Making a parametric assumption bears the risk of misspecification of
the distribution. The other approach is nonparametric, where there is no restriction on the number of parameters used by the candidate distribution family and the estimation is driven entirely by data. Without imposing a parametric assumption, the nonparametric approach is flexible and adaptive and avoids any distributional misspecification. This is the focus of the study presented in this thesis.

Of all nonparametric methods to density estimation, histograms are simple and easy to construct. They are likely the oldest yet still widely used tool for graphical presentation of data. However, their discontinuous nature not only fails to capture the underline smooth density, but also loses, to a significant extent, estimation accuracy. The less used frequency polygons provide continuous density estimates, by using a piecewise linear function that interpolates the midpoints of a histogram (Scott, 1992), but they still fail to capture smoothness. The kernel density estimator (KDE) is the most famous and theoretically well studied. The kernel density function is constructed by using the empirical mass function yet with each point-mass replaced with a continuous, symmetric distribution centered at each observation. It provides smooth density estimates, as long as the kernel used is smooth. The difficult part for the KDE is how to choose an appropriate value for the smoothing parameter, or bandwidth; see Jones et al. (1996) for a review of bandwidth selection methods. However, the KDE, being a convolution between the empirical mass function and the kernel function, tends to produce a flattened density estimate and hence gives a high bias and a low rate of convergence.

There also exist other nonparametric methods to density estimation, e.g., those under shape restrictions. These include Dumbgen and Rufibach (2009) for a smoothed version of the log-concave density estimator, Groeneboom et al. (2001), and Balabdaoui and Wellner (2010) for a piecewise linear decreasing and convex density estimator, and Papp and Alizadeh (2014) for a nonnegative polynomial splines estimator. Monographs on nonparametric density estimation are available by Silverman (1986), Scott (1992), Simonoff (1996) and Eggermont and LaRiccia (2001).
Mixture models, which can be seen as a generalization of the kernel estimator, can be very useful for nonparametric density estimation. Compared to the KDE, they are simpler in terms of the number of components yet may provide more accurate estimation due to their nature of deconvolution, as to be discussed next.

1.2 Using Mixtures for Density Estimation

A mixture distribution is a convex combination of component distributions. They are quite useful for modeling heterogeneity in statistical problems such as cluster analysis, density estimation and discriminant analysis. It consists of three main subgroups: finite, nonparametric and semiparametric mixture models. A finite mixture model is a weighted sum of a finite number of component distributions, where each component represents a subpopulation. The weights or mixing proportions of components are provided by the so-called mixing distribution. When the mixing distribution is completely unspecified, the mixture distribution is nonparametric. By including an additional structural parameter that is common to all components, a nonparametric mixture becomes semiparametric. Titterington et al. (1985) and McLachlan and Peel (2000) are well-known monographs on finite mixture models, while Lindsay (1995) and Böhnning (2000) provide excellent and extensive coverages on nonparametric and semiparametric mixture models.

Mixture models can also be very useful for density estimation and in fact, if used properly, can be advantageous over the KDE. An MDE (abbreviation for mixture-based density estimator/estimate/estimation) has a much simpler mathematical representation, because a mixture estimate often consists of only a few components, as opposed to a KDE that keeps a record of all observations. Moreover, compared to the KDE, a well-fitted mixture estimate is less biased and may tend to give a higher rate of convergence. Mixture-based density estimation is mainly studied from the viewpoint of likelihood and via the approach of likelihood maximization. With this
approach, most studies available in the literature focus on using finite normal mixtures (McLachlan and Peel, 2000; Scott and Szewczyk, 2001; Fruhwirth-Schnatter, 2006). There are two kinds of finite normal mixtures. It is homoscedastic, if the component variances are identical, or heteroscedastic, if otherwise. However, several difficulties exist for applying finite normal mixtures to density estimation. In the homoscedastic case, there may exist many local maximizers of the log-likelihood function for a given number of components, and computationally speaking, it is not always easy to find the global one. In the heteroscedastic case, the global maximizer that has an infinite likelihood value is useless as an estimator, while the local maximizer that is supposed to be consistent is not well defined for a finite sample, not to mention how to find it.

In terms of applying nonparametric mixtures to density estimation, it is theoretically guaranteed to be able to find the global maximum of a likelihood function and can potentially provide estimators with higher accuracy than the approach based on finite mixtures. However, it is computationally difficult to compute the nonparametric maximum likelihood estimate (NPMLE) of a mixing distribution and there had been very little progress made on using nonparametric mixtures for density estimation. Wang (2007) proposed a fast algorithm to compute an NPMLE, and by making use of this algorithm, Wang and Chee (2012) and Chee and Wang (2012, 2013) applied nonparametric and semiparametric mixtures to density estimation with great success. They proposed a general framework for univariate nonparametric density estimation based on mixture models. In particular, the component standard deviation is used to control the smoothness of the density estimate, hence also called bandwidth. For each fixed value of the bandwidth, a density estimate is determined by nonparametric likelihood maximization and then a model selection criterion is used to determine an appropriate bandwidth value. They showed that the density estimators using nonparametric mixtures outperform the KDE. However, due to various difficulties, their work has been limited to the univariate space.
It is the main goal of this thesis to extend their work to multivariate spaces, as well as to exploit some potentials provided by this extension.

1.3 Multivariate Density Estimation and Its Usage

In the multivariate space, the true distributions for real-world data are often very complicated and any parametric assumptions can hardly hold true, far more so than in the univariate case. To demonstrate this, let us take a real-world data set as an example. The forest fires data, available from the Machine Learning Repository at the University of California at Irvine (Bache and Lichman, 2013), has 13 variables. Here let us use five variables of them, DC index, ISI index, temperature, relative humidity and wind speed, with their univariate histograms shown in Fig. 1.1. It looks like that a two- or three-parameter family, e.g., a normal or a skewed normal distribution, would be able to model each the univariate distribution reasonably well. The situation gets complicated pretty quickly as the dimensionality increases. The pairwise scatter plots for this data set are shown in Fig. 1.2. Even for bivariate data, it is often not so easy to satisfactorily assume a parametric distribution. In a higher dimensional space, it is not difficult to imagine the increasing complexity of the true distributions. This suggests that nonparametric methods would be much more relevant due to its adaptive nature than their use in the univariate case.

Nonparametric density estimation from multivariate observations has many practical applications. Duong and Hazelton (2005) estimated the joint distribution of the under-5 mortality rate and the average life expectancy at birth in developing countries, Zhang et al. (2006) investigated the highest density region about the epicenters of earthquake tremors, and Ahmed et al. (2007) used multivariate density estimation to help detect network anomalies. With the new tools to density estimation made available in this thesis, we expect many more applications defined in
higher dimensional spaces will be made feasible.

The KDE is predominantly used in the literature for the task of nonparametric multivariate density estimation. However, in the multivariate case, the bandwidth turns into a matrix, e.g., the covariance matrix, and its selection, via cross-validation
or some purpose-built method, becomes much more challenging in practice. Most of the univariate bandwidth selection techniques can be extended in a relatively straightforward way to the multivariate case but only under the condition that the bandwidth matrix is constrained to be a diagonal matrix (Sain et al., 1994; Wand and Jones, 1994). Selection of a full bandwidth matrix is far more challenging than of a diagonal matrix, since the orientation of kernel functions to the coordinate axes needs to be considered, which does not have a univariate analogue. Some progresses about selecting a full bandwidth matrix have been made gradually. Duong and Hazelton (2003) proposed a new version of the plug-in selector for full bandwidth matrices, and Duong and Hazelton (2005) generalized earlier work about cross-validation selection methods from diagonal to full bandwidth matrices. Nevertheless, as will be shown in Chapter 3, the performance of these estimators is still not entirely satisfactory, and they suffer greatly from the problem of computational complexity, which restricts their applications to spaces of a low dimensionality, say, < 6.

Mixture models provide flexibility and are potentially very useful for multivariate density estimation, but multi-dimensional spaces present far greater challenges for mixture-based density estimation. For finite mixtures, it may require many components and hence is difficult to use a model selection method to select an appropriate number of components. Also, it is numerically troublesome to fit a heteroscedastic mixture, since the covariance matrices can easily become, undesirably, singular. See Scott (1992) and Fraley and Raftery (2002) for some work on finite-mixture-based density estimation in the multivariate case. For nonparametric mixtures, computing the multivariate NPMLE of a mixing distribution remains a great challenge. Moreover, estimating the component covariance matrix, though beneficial for performance improvement, is very difficult. Due to these difficulties, one can not extend directly the mixture-based approach of Wang and Chee (2012) from the univariate to the multivariate case. In particular, it is computationally too expensive to update the parameters of the nonparametric mixing distribution by a grid search in a
high-dimensional space. It is also computationally infeasible to use cross-validation to select proper values for the whole component covariance matrix.

In this thesis, we wish to provide a general method to finding a multivariate NPMLE, and to extend the work of Wang and Chee (2012) to the multivariate case. We shall tackle the challenges described above and provide solutions that are both advantageous in performance and feasible in computation. Some of the new ideas are as follows. We use a volume parameter of the component covariance matrix to control the density smoothness, and maximize the likelihood function of all unknown parameters, including the covariance matrix but excepting the volume parameter. The value of the volume parameter, being a scalar, will be determined via a model selection criterion. By making use of a special property of the gradient function, a random, rather than deterministic grid, is employed to help locate the local maxima of the gradient function in each iteration. New efficient algorithms are proposed for finding the maximum likelihood estimates of normal mixtures under various restrictions on the covariance matrix. Empirical studies using simulated and real-world data show that the new multivariate mixture-based density estimator performs remarkably better than kernel-based density estimators.

Successful multivariate density estimation can help solve many other statistical problems. In particular, we apply it to help solve two different types of problems. One is the classification problem. This is achieved by estimating nonparametrically the density function for observations in each class and classifying a new observation by the maximum a posterior criterion. Some specific issues of classification are also studied and discussed. Numerical studies show that the new classifier performs extremely well as compared with other commonly-used classification methods.

Another problem is to fit generalized linear mixed models with multiple random effects yet not following a parametric distribution. We study logistic regression models in detail and estimate nonparametrically the distribution of the random effects, along with the fixed effects. Simulation studies show that the new method
outperforms significantly the parametric method, in terms of both estimating the random and fixed effects, in situations when covariates are highly correlated and random effects follow a non-normal distribution. Applications to real-world data demonstrate that, compared to the traditional parametric methods, the new method obtains better estimates of the random effects, in terms of the likelihood values.

1.4 Contributions

The main contributions of the thesis are listed as follows:

- A new method for nonparametric multivariate density estimation which is based on nonparametric and semiparametric mixture distributions
- New efficient algorithms for finding the maximum likelihood estimates of multivariate nonparametric and semiparametric normal mixtures
- A new density-based classification method that uses multivariate semiparametric mixtures
- A quick method for variable elimination based on consideration of consistent density estimation for density-based classification
- Investigation of the influence of the complexity of discriminate boundaries on the accuracy of different classification methods
- A new method to fit generalized linear mixed models by using semiparametric mixtures to estimate the distributions of multiple random effects
- A new hybrid algorithm for fitting the generalized linear mixed models using multivariate semiparametric mixtures
- Investigation of the influence of the latent distribution of random effects on the performance of different estimation methods
1.5 Outline

The rest of the thesis is organized as follows. Chapter 2 provides a literature review on kernel density estimation, mixture models and mixture-based density estimation. Also, a brief overview is provided of multivariate density estimation and the relevant work about modeling with nonparametric density estimation. A short description of algorithms for fitting mixture models is given.

We systematically study nonparametric multivariate density estimation using mixtures in Chapter 3. A new method is proposed for computing the multivariate NPMLE of a mixing distribution for a nonparametric mixture model. Based on this work, a general approach is described for applying multivariate nonparametric and semiparametric mixtures to density estimation. Efficient algorithms for fitting the models in the multivariate case are proposed.

Chapter 4 carries out numerical studies, using simulated and real data, to compare the mixture-based method and the KDE.

Chapter 5 discusses classification problems based on density estimation. A new method that applies semiparametric mixtures to classification is described. Particularly, one semiparametric mixture distribution is used to provide a density estimate for each class, and smooth discriminant boundaries are naturally made available. Simulation studies using three different models are presented and six real-world data sets are analyzed.

Chapter 6 considers the problems of generalized linear mixed models, especially logistic regression models. A new method is presented that uses a semiparametric multivariate mixture formulation for generalized linear mixed models. Efficient algorithms are outlined for finding the nonparametric MLE of the mixing distribution of random effects, as well as the MLE of fixed effects. The comparison between the proposed method and the direct search directed derivative method for fitting semiparametric mixtures are also carried out.
Chapter 7 reports some empirical results from simulation studies and a real-world data set that compare the performance of the mixture-based method with the parametric method under the normality assumption.

Chapter 8 summarizes the thesis and presents some ideas for future research.
Chapter 2

Literature Review

2.1 Kernel Density Estimation

The kernel approach is likely the most popular tool for nonparametric density estimation due to its simplicity and well-established theoretical properties. The essences of kernel smoothing with kernel-based density estimator date back to the work of Rosenblatt (1956) and Parzen (1962). See Silverman (1986), Scott (1992) and Li and Racine (2007) for overviews on inspirations and developments of the KDE. Given a random sample \(x_1, \ldots, x_n\), the basic KDE may be written as

\[
\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left( \frac{x - x_i}{h} \right) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i),
\]

where \(K_h(t) = K(t/h)/h\) is the kernel indexed by \(h\), a smoothing parameter that is also known as the bandwidth of the KDE. Here \(K\) being a standard, symmetric, continuous density function. The KDE can be seen as a sum of ‘bumps’ placed at the observations, where \(K\) determines the shape of the bumps and \(h\) the width.

The main difficulty with kernel-based density estimation (KDE) is the bandwidth selection. Most bandwidth selectors target the mean integrated squared error:
\[ \text{MISE}(\hat{f}) = E \left\{ \text{ISE}(\hat{f}) \right\}, \quad (2.1) \]

where

\[ \text{ISE}(\hat{f}) = \int \{ \hat{f}(x) - f(x) \}^2 dx \]

\[ = \int \hat{f}(x)^2 dx - 2 \int \hat{f}(x)f(x) dx + \int f(x)^2 dx. \quad (2.2) \]

We note that the first term of (2.2) can be evaluated numerically and only depends on the density estimate, and that the third term does not depend on \( \hat{f} \) and hence can be ignored. By estimating the central term of (2.2), Rudemo (1982) and Bowman (1984) independently described the least squares cross-validation (CV) selector

\[ \text{CV}(h) = \int \hat{f}(x)^2 dx - \frac{2n^{-1}}{n} \sum_{i=1}^{n} \hat{f}_{-i}(x_i), \]

where \( \hat{f}_{-i}(x) \) is the kernel density estimate with the removal of the \( i \)th observation.

Inspired from their work, Scott and Terrell (1987) proposed the biased CV method and Hall et al. (1992) developed the smoothed CV method. Recently, Savchuk et al. (2010) presented the indirect CV method and Mammen et al. (2011) proposed the do-validation method by mingling different selectors.

Plug-in selectors approximate the mean integrated squared error by using bias and variance approximations,

\[ \text{MISE}(h) = \frac{p_0(K)^2 h^4}{4} \int f''(x)^2 dx + \frac{p_1(K)}{nh} + o\left\{ \frac{1}{nh} + h^4 \right\}, \quad (2.3) \]

where \( p_0(K) = \int v^2 K(v) dv \) and \( p_1(K) = \int K(v)^2 dv \). Minimizing (2.3) gives the
following asymptotically optimal bandwidth selector,

\[ h_{opt} = \left( \frac{p_1(K)}{np_0(K)^2 \int f''(x)^2 dx} \right)^{1/5}. \]

To estimate \( h_{opt} \), an estimate of \( \int f''(x)^2 dx \) needs to be substituted, which is usually derived from a “pilot” kernel density estimate. Compared to the CV methods, plug-in methods enjoy a faster convergence rate but their performance can be easily affected by the choice of pilots. The simplest plug-in selector is the rule of thumb method, based on normality assumption (Silverman, 1986). Sheather and Jones (1991) refined this selector criterion by improving the selection of pilots and later Chacon et al. (2008) provided data-dependent pilot bandwidth selectors.

All these methods above are called fixed kernel-based methods, where a global fixed bandwidth value is applied to all observations. Using a fixed bandwidth can cause some problems, e.g., underestimating the peaks and overestimating the valleys of the true density. To solve some of the problems caused by fixed bandwidth selectors, several improvements and modifications are introduced. For example, Breiman et al. (1977) and Abramson (1982) presented variable bandwidth kernel-based density estimators, and Choi and Hall (1999) and Hall and Minnotte (2002) proposed and studied a data sharpened kernel-based density estimator; Hall and Turlach (1999) and Hazelton and Turlach (2007) developed a weighted kernel-based density estimator.

### 2.2 Mixture Models

#### Finite Mixtures

Mixture models were first formally used 120 years ago by Pearson (1894), where a mixture of two univariate normal components was fitted to estimate the distribution of the ratio between size of forehead and body length of crabs. With a given number
of components, a finite mixture model has a density of the form

\[ f(x) = \sum_{j=1}^{m} \pi_j f_j(x), \]

where \( \pi_j > 0 \) and \( \sum_{j=1}^{m} \pi_j = 1 \). Here, \( f_j(x) \) and \( \pi_j \) denote the density function and mixing proportion for the \( j \)th component, respectively.

Practically, the normal family is often used for component distributions, which leads to normal mixtures (McLachlan and Peel, 2000; Fraley and Raftery, 2002). Other types of mixture models are also very useful; see, e.g., Peel and McLachlan (2000) for mixtures of \( t \)-distributions, and Wiper et al. (2001) for mixtures of Gamma distributions. Bouguila and ElGuebaly (2009) pointed out that the majority of references about finite mixture modeling focus on continuous data. For mixtures for discrete data, we refer to Vardi et al. (1985) and Church and Gale (1985) for Poisson, Juan and Vidal (2002) for Bernoulli, Chen et al. (2006) for multinomial and Melnykov (2013) for bell-shaped discrete mixture components.

**Nonparametric Mixtures**

In this thesis, we are mainly interested in nonparametric and semiparametric mixture models. A nonparametric mixture model has density of the form

\[ f(x; G) = \int_{\Omega} f(x; \theta) \, dG(\theta), \]  

(2.4)

where \( f(x; \theta), x \in \mathbb{R}^d, \theta \in \mathbb{R}^d \), is the component density function and \( G(\theta) \) the mixing distribution function with a completely unspecified form. According to Laird (1978) and Lindsay (1983), there always exists a discrete nonparametric maximum likelihood estimate (NPMLE) \( \hat{G} \) of \( G \) with no more support points than the number
of distinct values in the sample. A discrete $G$ can be written as

$$G(\theta) = \sum_{j=1}^{m} \pi_j \delta_{\theta_j},$$

where $\pi_j > 0$ for $j = 1, \ldots, m$, $\sum_{j=1}^{m} \pi_j = 1$, and $\delta_{\theta_j}$ puts mass 1 at $\theta_j$. For such a $G$, density (2.4) can be rewritten as

$$f(x; \pi, \theta) = \sum_{j=1}^{m} \pi_j f(x; \theta_j),$$

where $\pi = (\pi_1, \ldots, \pi_m)^T$, $\theta = (\theta_1, \ldots, \theta_m)^T$. With the likelihood approach used, one only needs to be concerned with discrete mixing distributions.

**Semiparametric Mixtures**

Semiparametric mixtures were first studied by Kiefer and Wolfowitz (1956). They have an additional finite-dimensional parameter $\beta$ which is common to all components. A semiparametric mixture model has density of the form

$$f(x; G, \beta) = \int_{\Omega} f(x; \theta, \beta) \, dG(\theta).$$

(2.5)

See Lindsay and Lesperance (1995) for a review on semiparametric mixture models.

**2.3 Maximum Likelihood Estimation of Mixture Models**

**Finite Mixtures**

The traditional method to estimating the parameters of a finite mixture model is maximum likelihood estimation via the expectation-maximization (EM) algorithm. For a heteroscedastic finite mixture of normal, letting $f$ be a normal density, and
\[ \pi = (\pi_1, \ldots, \pi_m)^T, \ \theta = (\theta_1, \ldots, \theta_m)^T \text{ and } \sigma = (\sigma_1, \ldots, \sigma_m)^T, \]

the log-likelihood function is

\[ l(\pi, \theta, \sigma) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} \pi_j f(x_i; \theta_j, \sigma_j) \right\}. \]

With unequal variances, \( l(\pi, \theta, \sigma) \) is not bounded and hence the MLE does not exist (Kiefer and Wolfowitz, 1956). However, for the likelihood equation, it has a sequence of consistent, asymptotically normal and efficient roots (Kiefer, 1978). With probability tending to one, these roots correspond to local maxima of \( l(\pi, \theta, \sigma) \). For a heteroscedastic mixture, the likelihood equation may have multiple roots, and it is generally impossible to decide which root is related to the consistent sequence.

The MLE of \((\pi, \theta, \sigma)\) for a homoscedastic finite mixture does exist. Redner (1981) pointed out that the MLE of homoscedastic finite mixtures is strongly consistent in the non-identifiable case. The consistency and asymptotic normality in identifiable situations by an unrestricted MLE was studied by Feng and McCulloch (1992). Some other results about the consistency problems are given by Peters and Walker (1978), Cheng and Liu (2001) and Atienza et al. (2007). However, the likelihood function of a finite mixture may have many local maxima. Therefore, it may not be easy to find the global maximum of the likelihood function when computing the MLE.

**Nonparametric Mixtures**

Maximum likelihood is the most popular approach to finding an estimate of \( G \) for nonparametric mixture models. Other methods are discussed by, e.g., Prakasa Rao (1983), chapter 10.

Theories of the nonparametric maximum likelihood estimate (NPMLE) have been studied for a long time. Particularly, Kiefer and Wolfowitz (1956) studied the consistency of the NPMLE under certain regularity conditions, Laird (1978) proved...
that the NPMLE is self-consistent and characterized by a step function in the non-
identically distributed case, and Lindsay (1983) systemically displayed the character-
ization of the NPMLE in the non-identically distributed case by using convex
geometric analysis.

Given a random sample $x_1, \ldots, x_n$, the log-likelihood function of the nonparamet-
metric mixing distribution is given by

$$l(G) = \sum_{i=1}^{n} \log \left\{ \int_{\Omega} f(x; \theta) \ dG(\theta) \right\}.$$  

For a discrete $G$, the log-likelihood function is thus

$$l(G) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} \pi_j f(x_i; \theta_j) \right\}.$$  

The nonparametric maximum likelihood estimate $\hat{G}$ can be characterized by the
gradient function, which is a special directional derivative of the log-likelihood func-
tion. It is defined as

$$d(\theta; G) = \frac{\partial l\{(1 - \varepsilon)G + \varepsilon \delta_{\theta}\}}{\partial \varepsilon} \bigg|_{\varepsilon=0}$$

$$= \sum_{i=1}^{n} \frac{f(x_i; \theta)}{f(x_i; G)} - n. \quad (2.6)$$

It characterizes $\hat{G}$, since

$$\hat{G} \text{ maximises } l(G) \iff \hat{G} \text{ minimizes } \sup_{\theta} \left\{ d(\theta; G) \right\} \iff \sup_{\theta} \left\{ d(\theta; \hat{G}) \right\} = 0.$$
Semiparametric Mixtures

For semiparametric mixture models, since there must exist a discrete NPMLE for every fixed $\beta$, the log-likelihood function can be written as

$$l(G, \beta) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} \pi_j f(x_i; \theta_j, \beta) \right\}.$$  \hspace{1cm} (2.7)

Maximizing the log-likelihood function (2.7) gives the semiparametric maximum likelihood estimate $(\hat{G}, \hat{\beta})$.

For any given $\beta$, the gradient function for the semiparametric mixture model is defined by

$$d(\theta; G, \beta) \equiv \frac{\partial l\{(1 - \varepsilon)G + \varepsilon\delta_{\theta}, \beta\}}{\partial \varepsilon} \bigg|_{\varepsilon=0} = \sum_{i=1}^{n} f(x_i; \theta, \beta) f(x_i; G, \beta) - n.$$  \hspace{1cm} (2.8)

For $(\hat{\beta}, \hat{G})$ to be the semiparametric MLE, there are two necessary conditions. One is $\sup_{\theta} \{d(\theta; \hat{G}, \hat{\beta})\} = 0$, and the other

$$\left( \frac{\partial l}{\partial \beta} \right) (\hat{G}, \hat{\beta}) = 0.$$

2.4 Mixture-based Density Estimation

Density estimation using nonparametric mixture models is mostly restricted to using normal location mixtures, and model fitting is usually done by maximum likelihood estimation. However, a density function can not be estimated completely nonparametrically by the maximum likelihood method, if no restriction is enforced on the mixing distribution or the scale parameter of the normal distribution. This is because the parameter space is too large and the global maximum of the likelihood function does not exist. To overcome this, Geman and Hwang (1982) considered
applying Grenander’s method of sieves (Grenander, 1981) to create a constrained subspace of the parameter space (called a sieve) and maximizing the likelihood over such a subspace. In this context, the subspaces of the parameter space are called the normal convolution sieve. They considered two variants of normal convolution sieve to reduce the computational complexity. Particularly, they firstly defined a constrained subspace of the parameter space and then relax the constraint as the sample size grows. However, with such variants, the models become finite mixtures. See Jones and Henderson (2009) for further studies of the variants. Following German and Hwang (1982), Roeder (1990, 1992) applied a normal location mixture to density estimation and the method of spacings to estimate the mixing distribution. By treating the scale parameter as a sieve, the mixing distribution can be estimated by maximizing the product of spacings with the scale parameter held fixed, and a confidence band of density estimates can be constructed. For the density estimate itself, an appropriate scale parameter can be obtained by the cross-validation method.

Other attempts at fitting nonparametric mixtures to discrete data include, but are not limited to, Simar (1976) for the Poisson distribution, Jewell (1982) for the exponential distribution and Böhning and Patilea (2005) for general power series distributions including Poisson, negative binomial and logarithmic series as well as binomial.

A highly relevant work to our studies is Wang and Chee (2012). They investigated a nonparametric mixture-based density estimator and named it the mixture-based density estimator (MDE), which is given by

$$f_h(x; \pi, \theta) = \sum_{j=1}^{m} \pi_j f_h(x; \theta_j),$$

where $\theta = (\theta_1, \ldots, \theta_m)^T \in \mathbb{R}^m$ and $\pi = (\pi_1, \ldots, \pi_m)^T$, with $\pi_j > 0$ for $j = 1, \ldots, m$, $\sum_{j=1}^{m} \pi_j = 1$. It is virtually of the same form as the KDE but more general, since
the support points $\theta_j$ can be different from observations and weights assigned to the components, $\pi_j$, are not fixed at $1/n$. The number of components $m$ is often far smaller than $n$ and is automatically determined by maximum likelihood for any fixed value of $h$. To apply the MDE to nonparametric density estimation, one can not simply maximize the likelihood function with $h$ included as an argument. This is because the profile log-likelihood $\tilde{l}(h) \equiv l_h(\hat{G}_h) \equiv \max_G l_h(G)$ approaches $\infty$ as $h \to 0$, with $\hat{G}_h$ approaches the empirical distribution function. This is not a desired solution to smooth density estimation. To overcome this, Wang and Chee (2012) proposed to find the maximum likelihood estimate $\hat{G}_h$ (and $\hat{\beta}_h$, in the semiparametric case) for each $h$-value defined on a grid which should exclude $h$-values near 0. By using a model selection criterion, e.g., the AIC, a proper value out of all candidate $h$s and thus the corresponding $\hat{G}_h$ (and $\hat{\beta}_h$) can be decided.

2.5 Multivariate Density Estimation

In the multivariate case, we focus on kernel-based and mixture-based density estimation. Given a $d$-variate random sample $x_1, \ldots, x_n$, a multivariate kernel density estimator is defined by 

$$\hat{f}(x; H) = \frac{1}{n} \sum_{i=1}^{n} K_H(x - x_i),$$

where $x = (x_1, \ldots, x_d)^T$ and $x_i = (x_{i1}, \ldots, x_{id})^T$, $i = 1, \ldots, n$. Here $K(x)$ is the multivariate kernel function which is a standard, symmetric, continuous density function, and $K_H(x) = |H|^{-\frac{1}{2}} K(H^{-\frac{1}{2}} x)$, where $H$ is the bandwidth matrix. Since the bandwidth parameter of the KDE becomes a matrix, more difficulties are added to its selection procedure. Many research efforts have been made on it. Wand and Jones (1994) extended the univariate plug-in selector of Sheather and Jones (1991) to the multivariate case, under the restriction that the bandwidth matrix be di-
agonal. Sain et al. (1994) derived a multivariate version of the univariate biased cross-validation selector of Scott and Terrell (1987), also enforcing the diagonality restriction. As noted by Sain et al. (1994) and Wand and Jones (1994), using only diagonal bandwidth matrices may lead to remarkably suboptimal density estimates. To overcome it, Duong and Hazelton (2003) proposed a new type of plug-in selector which always produces a full bandwidth matrix. Duong and Hazelton (2005) presented new cross-validation selectors for selecting full bandwidth matrices, but it seems that their performance depends heavily on pilot bandwidth matrices. Duong (2008) carried out simulation studies to compare some of these bandwidth selectors and found that the last two selectors that use full bandwidth matrices had better performance.

Literature about density estimation using mixture models mostly focuses on using finite mixture (Scott, 1992; McLachlan and Peel, 2000; Scott and Szewczyk, 2001; Fraley and Raftery, 2002). A finite normal MDE is given by,

$$f(x) = \sum_{j=1}^{m} \pi_j \phi_j(x|\theta_j, \Sigma_j),$$

where $\pi_j$ is the proportion of the $j$th mixture component and $\phi_j(x|\theta_j, \Sigma_j)$ the normal density for the $j$th component with mean $\theta_j$ and covariance matrix $\Sigma_j$. The maximum likelihood estimate of the mixture is usually computed for each fixed number of mixture components and a model selection method is used to determine a proper number of components. However, finite mixture estimation is challenging due to the fact that the likelihood function of a homoscedastic mixture has several local maxima while the likelihood function of a heteroscedastic mixture is unbounded. Moreover, given a finite sample, there is no well defined local maximizer of the likelihood function that is consistent for a heteroscedastic mixture.

There has been very little progress made on nonparametric MDE because of various challenges. In fact, computing the multivariate NPMLE of a mixing distribution
has not been well solved. Pilla et al. (2006) tried to extend the work of estimating an univariate NPMLE to the multivariate case, with a focus on multivariate normal mixtures. They used a sieve parameter to control the smoothness of the density estimates, and estimated the mixing distribution with a fixed sieve parameter value. The component covariance matrix they used is simply the sample covariance matrix multiplied by the sieve parameter. Nor did they not address the problem of the choice of the sieve parameter. Chung and Lindsay (2011) tried to apply non-parametric mixtures to density estimation by treating the kernel density estimator as a component of the nonparametric mixture model and improving its likelihood with the EM algorithm. They mentioned that although their work focuses on the univariate case, it may be extended to the multivariate case. Among these works, the selection of the entire bandwidth matrix is completely untouched.

### 2.6 Usage of Density Estimation

Density estimation is a fundamental problem in statistics and its solution provides a basic tool to solve other types of statistical problems. Therefore, by making use of successful density estimation methods, improvements can be made for solving other statistical problems, e.g., classification analysis, generalized linear mixed models, cluster analysis and regression analysis. This section discusses briefly two types of problems that may rely on density/distribution estimation, which will be later investigated in the thesis. In the discussion below, we focus on how such problems can be solved based on density estimation.

**Classification**

Classification is often solved by identifying the specific data features for each given category and matching the observation with the category that has the most similar features with it. Here a category is also called a class and the procedure a classifier.
There is a vast number of practical classification applications. Aram et al. (2013) investigated the late osteolysis after cemented Charnley total hip arthroplasty, Patel and Choi (2014) studied the reliability assessment of structural systems, and Sundaramoorthi (2014) were interested in forecasting ground-level ozone concentration levels in order to reduce its hazard to people’s health and the environment.

Many classification methods are based on density estimation, which first estimate a density function for each class and label an observation using the maximum a posterior (MAP) criterion, i.e., as the class with the highest posterior probability. The posterior probability for an observation \((y, \mathbf{x})\) belonging to class \(k\) is given by

\[
\Pr[y = k | \mathbf{x}] = \frac{p_k f_k(\mathbf{x})}{\sum_{l=1}^{q} p_l f_l(\mathbf{x})},
\]

where \(f_k(\mathbf{x})\) denotes the density function for an observation in class \(k\) \((=1,\ldots,q)\) and \(p_k\) the proportion of observations of class \(k\) in the population.

A crude method is to use a simple distribution, e.g., a normal distribution, for modeling the subpopulation in each classes. One of the earliest work is linear discriminant analysis, which was first applied by Barnard (1935) following the suggestion of Fisher and formally investigated later by Fisher (1936). It can be derived based on the assumptions that the observations in each class follow a multivariate normal distribution and the covariance matrices of different classes are identical. It leads to quadratic discriminant analysis if non-identical covariance matrices are used (see, e.g., Fraley and Raftery (2002) and Hastie et al. (2009)). However, normality assumptions can only give simple boundaries and may become very inappropriate in a complicated situation. In such cases, nonparametric density estimators would be able to offer better solutions. Of course, kernel density estimation can be used off the shelf to estimate each class density, and this is known as kernel discriminant analysis. This generalizes linear and quadratic discriminant analysis by using the idea of density estimation that automatically defines discriminant boundaries.
Mixture models have also been applied to classification, sometimes known as model-based discriminant analysis (MDA), but so far their use is only limited to finite mixtures (Fraley and Raftery, 2002; Hastie et al., 2009). Hastie and Tibshirani (1996) studied using finite mixtures with identical covariance matrices within each class. Bensmail and Celeux (1996) considered the eigenvalue decomposition of covariance matrices, defined subfamilies of finite mixtures and applied them to MDA. Compared to traditional linear and quadratic discriminant analysis, the MDA is more flexible and suitable for classes with complicated boundaries.

The performance of a classifier based on density estimation depends, to a great extent, on that of the density estimator used. The problems that are associated with the KDE and finite mixtures as described earlier in this chapter, naturally carry over when they are used for classification purposes.

### A Special Mixed Effects Models

Generalized linear mixed models (GLMM’s) extend generalized linear models by including random coefficients in the linear predictor. They are quite useful for repeated measurements and clustered data. For clustered data, let us assume that there are \( n_i \) observations from the \( i \)th cluster, \( i = 1, \ldots, n \), and that the \( j \)th observation of cluster \( i \) has random-effects covariates \( x_{ij} \) and fixed-effects covariates \( z_{ij} \).

Let the conditional joint density of \( \mathbf{y}_i = (y_{i1}, \ldots, y_{in_i})^T \) be given by

\[
 f(\mathbf{y}_i| \mathbf{X}_i, \mathbf{Z}_i, \mathbf{\theta}_i, \mathbf{\beta}) = \prod_{j=1}^{n_i} f(y_{ij}| x_{ij}, z_{ij}, \theta_i, \beta), \tag{2.10}
\]

where \( \mathbf{X}_i = (x_{i1}, \ldots, x_{in_i})^T \), \( \mathbf{Z}_i = (z_{i1}, \ldots, z_{in_i})^T \), and \( \theta_i \) and \( \beta \) are the random and fixed effects, respectively. Let

\[
 \mu_{ij} = E[y_{ij}| x_{ij}, z_{ij}, \theta_i, \beta]
\]
and for a linear predictor, the covariate vectors are connected to $\mu_{ij}$ through a link function

$$g(\mu_{ij}) = x_{ij}^T \theta_i + z_{ij}^T \beta.$$  \hfill (2.11)

One main interest in fitting the GLMM’s is to estimate the distribution function of random effects. Typically, random effects are assumed to follow a normal distribution (Laird and Ware, 1982; Breslow and Clayton, 1993; Hedder and Gibbons, 1994). The normality assumption can sometimes be convenient for analysis and computation and may possibly provide robust estimation of fixed effects (Neuhaus et al., 1992, 1994; Butler and Louis, 1997). However, the misspecification of the random effects distribution generally compromises the estimation efficiency (Tao et al., 1999).

To protect against misspecification of random effects distribution, mixture models with its flexibility are also used in GLMM’s. Stiratelli et al. (1984) proposed to use a finite mixture of normal distributions to approximate the random effects distribution in longitudinal studies. Verbecke and Lesaffre (1996) demonstrated that random effects could be badly estimated under the normality assumption if random effects are generated from a finite mixture of normal distributions; see McCulloch et al. (2008) for an extensive study.

Nonparametric distributions for random effects can also be used in GLMM’s, with the advantage of no distributional misspecification. Aitkin (1996) used the NPMLE to solve the problem of overdispersion in generalized linear models, which is very much the same problem as a random intercept in a GLMM, as studied by Aitkin (1999), Murphy and van der Vaart (2000) and Wang (2010). Recently, Lesperance et al. (2014) proposed to use the NPMLE to estimate the distribution of a random intercept and a random slope. However, due to the computational complexity, these nonparametric approaches are only applicable to the situation
with one or two random effects.

2.7 Algorithms for Fitting Mixture Models

The Expectation-maximization Algorithm

The expectation-maximization (EM) algorithm is very useful for computing maximum likelihood estimates from incomplete data and was formalized in the seminal paper of Dempster et al. (1977). In this paper, the expectation step (or E step) and the maximization step (M step) were recognized in their general forms, and some theoretical properties were established. Since then, the EM algorithm has been widely applied to fit finite mixture models when data are heterogeneous. A systematical coverage of the EM algorithm is given by McLachlan and Krishnan (1997).

The EM algorithm iterates between two steps, the E and M step, until it converges. In the E step, the expectation of the complete log-likelihood given the data and current parameter estimates over the missing data is computed. The expectation, denoted by a $Q$-function, is given by

$$Q(\Phi^*|\Phi) = \mathbb{E}(l_c(\Phi^*)|\Phi, x),$$

where $l_c(\Phi^*)$ denotes the complete log-likelihood function, and $\Phi^*$ the unknown parameters to be updated and $\Phi$ the current parameter estimates. Then in the M step, the parameters are updated by maximizing the $Q$-function, namely

$$\Phi' = \arg \max_{\Phi^*} Q(\Phi^*|\Phi).$$
For a homoscedastic finite normal mixture, the log-likelihood function is given by

\[ l(\pi, \Theta, \Sigma) = \sum_{i=1}^{n} \log \left\{ f(x_i; \pi, \Theta, \Sigma) \right\}, \]

where \( \pi = (\pi_1, \ldots, \pi_m)^\top \), \( \Theta = (\theta_1, \ldots, \theta_m) \), and \( \theta_j \) and \( \Sigma_j \) are the mean vector and covariance matrix for the \( j \)th component distribution. To update from \((\pi, \Theta, \Sigma)\), the EM iteration formulae are given by

\[
\begin{align*}
\pi_j' &= \frac{1}{n} \sum_{i=1}^{n} p_{ij}, \\
\theta_j' &= \frac{\sum_{i=1}^{n} p_{ij} x_i}{\sum_{i=1}^{n} p_{ij}}, \\
\Sigma_j' &= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} (x_i - \theta_j)(x_i - \theta_j)^\top
\end{align*}
\]

where

\[ p_{ij} = \frac{\pi_j f_{ij}}{\sum_{l=1}^{m} \pi_l f_{il}}, \]

and \( f_{ij} = f(x_i; \theta_j, \Sigma) \).

The EM algorithm has an easy implementation and is numerically stable. Its main disadvantage is slow convergence of a linear, and sometimes sublinear, order.

**The Constrained Newton Method**

Several methods were proposed to compute the NPMLE, although some were originally described in the context of optimal design (Silvey, 1980; Pukelsheim, 1993). Fedorov (1972), Wynn (1970) and Wu (1978) proposed the vertex direction method (VDM), followed by the EM algorithm (Laird, 1978) and Simar’s method (Simar, 1976; Böhning, 1982). Böhning (1985) developed the vertex exchange method (VEM), and Lesperance and Kalbfleisch (1992) proposed the intra-simplex direction
method (ISDM) and applied the semi-infinite programming (SIP) method of Coope and Watson (1985). Wang (2007) proposed the CNM (the constrained Newton method with multiple support points) algorithm which can be seen as an extension of the quadratic method of Atwood (1976).

Of these methods, only the EM algorithm chooses a large number of initial support points, while the others only add one or more new useful support points iteratively for finding \( \hat{G} \). The speed of convergence plays a significant role in an iterative computational method. Moreover, since the number of support points may expand explosively, techniques such as discarding information of support points in previous iterations, collapsing similar ones or discarding bad ones are necessary for reducing computational complexity. Therefore, many methods mentioned above may converge too slowly to be useful in some applications.

Wang (2007) proposed the constrained Newton method with multiple support points (CNM) for computing an NPMLE, which has been shown to be very fast and stable, and appears to be the most efficient for finding an accurate NPMLE. The CNM method consists of two parts: updating \( \pi \) with \( \theta \) fixed, and updating \( \theta \). To update \( \pi \) with fixed \( \theta \), first we have

\[
\frac{\partial l}{\partial \pi} = S^T 1,
\]

\[
\frac{\partial^2 l}{\partial \pi \partial \pi^\top} = -S^T S,
\]

where \( s_i(\pi, \theta) = \frac{\partial \log(f(x_i; \pi, \theta))}{\partial \pi} \), \( i = 1, \ldots, n \), \( S = (s_1(\pi, \theta), \ldots, s_n(\pi, \theta))^\top \) and \( 1 = (1, \ldots, 1)^\top \). The Taylor series expansion about \( \pi \) gives

\[
l(\pi, \theta) - l(\pi', \theta)
\approx -1^\top S(\pi' - \pi) + \frac{1}{2}(\pi' - \pi)S^\top S(\pi' - \pi)
\]

\[
= \frac{1}{2} \| S\pi' - 2 \|^2 - \frac{n}{2}.
\]
where $\mathbf{2} = (2, \ldots, 2)^T$ and $\|\cdot\|$ denotes the $L_2$-norm. Therefore, to maximize $l$ over $\pi$ with $\theta$ fixed, one can iteratively solve the following least squares linear regression problem over a probability simplex:

$$\min_{\pi'} \| \mathbf{S}\pi' - \mathbf{2} \|^2, \text{ subject to } \pi'^T \mathbf{1} = 1, \pi' \geq 0. \quad (2.12)$$

The unity constraint can be removed by a transformation suggested by Dax (1990), which works as follows. If $\tilde{\pi}$ solves

$$\min_{\pi} \| \mathbf{P}\tilde{\pi} \|^2 + |\tilde{\pi}^T \mathbf{1} - 1|^2, \text{ subject to } \tilde{\pi} \geq 0, \quad (2.13)$$

where $\mathbf{P} \equiv (\mathbf{s}^{(1)} - \mathbf{2}, \ldots, \mathbf{s}^{(m)} - \mathbf{2})$, $\mathbf{s}^{(j)}$ being the $j$th column of $\mathbf{S}$, then $\tilde{\pi}/\tilde{\pi}^T \mathbf{1}$ solves problem (2.12). The transformed problem (2.13) is easily solvable by the non-negative least squares (NNLS) algorithm of Lawson and Hanson (1974), p.161.

To iteratively update $\theta$, the CNM firstly expands the support points set by including all the local maxima of the gradient function (2.6), and then after updating $\pi$, shrinks the support point set by discarding those elements with mass zero. The CNM algorithm can be simply described as follows.

**Algorithm 1 (CNM)** Set $t = 0$, and choose $G_0$ with a finite support such that $l(G_0) > -\infty$. Repeat the following steps.

**Step1:** compute all local maxima $\theta^*_{t1}, \ldots, \theta^*_{tn_g}$ of the gradient function (2.6), $\theta \in \Theta$. If $\max_{j=1, \ldots, n_g} \{d(\theta^*_{tj}; G_t)\} = 0$, stop.

**Step2:** set $\theta^+_t = (\theta^T_t, \theta^*_{t1}, \ldots, \theta^*_{tn_g})^T$ and $\pi^+ = (\pi^T_t, \mathbf{0}^T)^T$. Find $\pi^-_{t+1}$ by solving problem (2.12), followed with a line search.

**Step3:** Discard all support points with zero entries in $\pi^-_{t+1}$, which gives $G_{t+1}$ with $\theta_{t+1}$ and $\pi_{t+1}$. 

An Algorithm for fitting Semiparametric Mixtures

To find \( \hat{G} \) and \( \hat{\beta} \) jointly in the case of a semiparametric mixture, Wang (2010) proposed three general algorithms, all combining the CNM with an unconstrained optimization algorithm. Of the three, the CNM-MS algorithm (short for constrained Newton method by modifying the support) seems to be the most efficient. It iterates between updating only \( G \) with one iteration of the CNM, thus allowing for the expansion of the support set and giving a \( \pi \) that is interior to the probability simplex, and solving the optimization problem over the finite-dimensional parameter vector \((\pi^T, \theta^T, \beta^T)^T\) with a commonly-used optimization algorithm, e.g., the quasi-Newton BFGS (Broyden-Fletcher-Goldfarb-Shanno) method. Here \( \pi_- \) denotes \( \pi \) less one element (either the last or the largest element), as a result from removing the equality constraint \( \pi^T 1 = 1 \). Note that when updating \((\pi_-^T, \theta^T, \beta^T)^T\), some elements of \( \pi \) may become zero and subsequently removed, and hence the estimate of \( \pi \) may have a reduced dimension at the end of the unconstrained optimization step. The outline of the CNM-MS can be described as follow.

Algorithm 2 (CNM-MS). Set \( t = 0 \), and choose \( \beta_0 \) and \( G_0 \) with a finite support such that \( l(G_0, \beta_0) > -\infty \). Repeat the following steps.

Step1: update \((\pi_t, \theta_t)\) to \((\pi_{t+\frac{1}{2}}, \theta_{t+\frac{1}{2}})\) with \( \beta = \beta_t \) fixed by running one iteration of the CNM method.

Step2: update \((\pi_{t+\frac{1}{2}}, \theta_{t+\frac{1}{2}}, \beta_t)\) to a local maximum \((\pi_{t+1}, \theta_{t+1}, \beta_{t+1})\) by using the BFGS method with a line search.

Step3: set \( s = s + 1 \). Stop if converged.
Chapter 3

Nonparametric Multivariate Density Estimation Using Mixtures

3.1 Introduction

The aim of this chapter is to extend properly the nonparametric/semiparametric MDE approach of Wang and Chee (2012) to the multivariate case. A major difficulty with applying this MDE approach in the multivariate case is how to choose a bandwidth (or component covariance) matrix. Leaving it completely unspecified gives useless degenerate estimated components that are located at observations, as the likelihood is infinite in this case (Grenander, 1981; Geman and Hwang, 1982). On the other hand, selecting proper variance and covariance values for the whole matrix via, say, cross-validation is computationally infeasible. To overcome this, we propose to select, via the Akaike information criterion (AIC) (Akaike, 1974), only the value of a volume parameter of the bandwidth matrix and leave the rest of the matrix to be determined by likelihood maximization. This successfully bounds the likelihood function without leaving too many parameter values for selection. As we shall see, this volume parameter in the univariate case is just the scalar bandwidth, which makes the proposed extension methodologically coherent. Computational
algorithms are also proposed for finding the corresponding estimates.

The remainder of this chapter is organized as follows. Section 3.2 describes multivariate nonparametric and semiparametric mixture models and how they can be applied in a practically feasible way to multivariate density estimation. Section 3.3 proposes efficient algorithms for fitting the models in the multivariate case. Some discussion are provided in Section 3.4.

3.2 Density Estimation using Multivariate 
Nonparametric/Semiparametric mixtures

3.2.1 Nonparametric formulation

For multivariate observations, when the component covariance matrix $H$ is held fixed, the nonparametric mixture density is given by

$$f_H(x; G) = \int_{\Omega} f(x; \theta, H) \, dG(\theta),$$

(3.1)

where $x \in \mathbb{R}^d$, $\theta \in \Omega \subset \mathbb{R}^d$ and $f(x; \theta, H)$ is the component density centered at $\theta$ and with covariance matrix $H$. Throughout this chapter, we focus on using the multivariate normal density for $f(x; \theta, H)$, although it does not have to be in most cases, and may simply call $H$ the bandwidth matrix as it plays the same role in controlling the smoothness of the density estimate as for the KDE. For a discrete $G$ supported at $\theta_1, \ldots, \theta_m$ with, respectively, masses $\pi_1, \ldots, \pi_m$, one can rewrite density (3.1) as

$$f_H(x; \pi, \Theta) = \sum_{j=1}^{m} \pi_j \, f(x; \theta_j, H),$$
where $\pi = (\pi_1, \ldots, \pi_m)^\top$ and $\Theta = (\theta_1, \ldots, \theta_m)$. Since only discrete $G$s need to be considered in this chapter, we often treat $G$ and its $(\pi, \Theta)$ interchangeably. The log-likelihood function with $H$ fixed is given by

$$l_H(G) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} \pi_j f(x_i; \theta_j, H) \right\}. \quad (3.2)$$

The likelihood theory remains the same as in the univariate case which is reviewed in Chapter 2.3, including the theoretical results that concern the gradient function, here given by

$$d_H(\theta; G) = \sum_{i=1}^{n} \frac{f(x_i; \theta, H)}{f_H(x_i; G)} - n. \quad (3.3)$$

As described in Chapter 2.3, the gradient function is a directional derivative of the log likelihood function. While adding the points with positive gradient values to support points set, the log-likelihood function will increase. And the point with the largest gradient value is included as new support point, the log-likelihood function increases most. The most efficient way to increase the log-likelihood function is to find all the local maxima of gradient function. The gradient function characterizes the NPMLE $\hat{G}$ according to the general equivalence theorem,

$$\hat{G} \text{ maximises } l_H(G) \iff \hat{G} \text{ minimizes } \sup_{\theta} \{d_H(\theta; G)\} \iff \sup_{\theta} \{d_H(\theta; \hat{G})\} = 0.$$

If there is no any point that can make the gradient function more than 0, it means we find the NPMLE. In the univariate space, the CNM (Wang, 2007) uses a deterministic grid to search all the local maxima of gradient function, but its computational complexity will increase exponentially with the dimensions. We shall describe how to compute the NPMLE $\hat{G}$ in the multivariate case in Section 3.3.3.

The maximum likelihood can not be directly used to estimate a multivariate nonparametric mixture. This is because we can not maximize the likelihood function...
with $H$ simply treated as an argument, since the likelihood will become infinite and $H$ singular. On the other hand, it will also be very difficult to apply the above formulation directly to nonparametric density estimation, as $H$ has $(d^2 + d)/2$ unknown elements. A selection method for the entire $H$ via cross-validation or model selection will quickly turn to be computationally intractable as $d$ increases. We need to reduce considerably the number of unknowns left for selection while keeping the likelihood bounded.

### 3.2.2 Semiparametric formulation

The simplest way, and perhaps the best for our purposes here, to bound the likelihood or the component density function is to introduce a volume parameter for $H$ and have its value held fixed. Consider the following decomposition of $H$:

$$H = h^2 B,$$

where $h = |H|^{\frac{1}{n}}$ and $B$ is a symmetric and positive-definite matrix satisfying $|B| = 1$. Here $h$ determines the volume of $H$, while $B$ the shape and orientation.

Now with $h$ fixed, the mixture density with a discrete $G$ becomes

$$f_h(x; G, B) = \sum_{j=1}^{m} \pi_j f(x; \theta_j, h^2 B)$$

and the log-likelihood

$$l_h(G, B) = \sum_{i=1}^{n} \log \left\{ f_h(x_i; G, B) \right\}. \quad (3.4)$$

Note that the mixture is now semiparametric, with $B$ being the parametric unknown, and that $B$ is restricted to be positive-definite. Since, being the normal density, $f(x; \theta_j, H) \leq (2\pi)^{-\frac{d}{2}} |H|^{-\frac{1}{2}}$ for all $x \in \mathbb{R}^d$ and all $\theta_j \in \Omega$, the log-likelihood is
bounded by

\[ l_h(G, B) \leq -\frac{nd}{2} \log(2\pi h^2). \]

The estimation procedure consists of two steps. The first step is to maximize log-likelihood (3.4) with respect to \( G \) and \( B \) for each \( h \)-value fixed, and the second is to select a proper \( h \)-value out of all candidate ones based on a model selection criterion. The role played by \( h \) here is exactly the same as by the bandwidth scalar in the univariate case. It controls the smoothness of the density estimate but leaves the shape and the orientation of a component to be determined by likelihood maximization. Since there is only a scalar value to be determined by selection, it is computationally feasible.

One can further impose various restrictions on \( B \), based on background knowledge or consideration of model complexity. For example, one may restrict \( B \) to be an identity matrix or diagonal. In these two cases, the numbers of free parameters in \( B \) are, respectively, 0 and \( d - 1 \), as opposed to \((d^2 + d)/2 - 1\) for a full \( B \). Such additional restrictions make very little difference in the computation method described in Section 3.3.2.

### 3.2.3 Volume selection

Through controlling \( h \) and profiling the likelihood function, a sequence of semiparametric mixtures is defined as a continuous family of candidate density estimates. Then by using a model selection criterion, an approximate \( h \) is chosen and thus the corresponding density estimate is produced. Since an interactive selection via visual examination is impossible in a high-dimensional space, one can resort to either simulation-based techniques, such as cross-validation and bootstrapping, or parsimony-based model selection criteria, such as the AIC or BIC (Bayesian information criterion) (Schwarz, 1978).
The use of maximum likelihood estimation in our procedure not only provides good density candidate estimates for different $h$-values but also benefits in using an information-theoretic model selection criterion to choose a proper value for $h$. In particular, we use the AIC because it appears to be more reliable than the BIC for mixture-based estimators (Wang and Chee, 2012). The AIC for the profile likelihood minimizes over $h$

$$\text{AIC}(h) = -2\tilde{l}(h) + 2p, \quad (3.5)$$

where $\tilde{l}(h) \equiv \max_{G, B} l_h(G, B)$ is the profile log-likelihood function of $h$ and $p$ the number of free parameters including $h$. For a $d$-dimensional data set, a $\hat{G}$ with $m$ components and a full $B$, we have $p = m(d + 1) + (d^2 + d)/2 - 1$.

However, there are some noteworthy problems associated with the use of likelihood-based target function. According to Kiefer and Wolfowitz (1956), the profile likelihood increases monotonically and approaches infinity as $h \to 0$, which means the AIC can not be applied to $h$ near zero. Practically, supplying a positive lower bound on $h$ can avert this problem. Furthermore, the asymptotic normal theory of maximum likelihood depends strongly on regularity conditions (boundary conditions and nested parameter structure among others), which are known to fail for mixture models (Lindsay, 1995). Although theoretical justifications are lacking in applying the AIC to mixture models, the AIC has been proven to perform reasonably well for mixture models in the univariate case (Wang and Chee, 2012). Our simulation and real-world data studies in the multivariate case have come to a similar conclusion.
3.3 Computation

3.3.1 A hybrid approach

With $h$ fixed, one needs to maximize the log-likelihood function (3.4) with respect to $G$ and $B$. The nonparametric $G$ and the positive definiteness restriction on $B$ make it very difficult to use a single optimization algorithm to find the MLE $(\hat{G}_h, \hat{B}_h)$. Furthermore, for a given number of support points of $G$, we want to take the advantage that it is a finite mixture and can be fitted more directly and rapidly. Hence a hybrid approach seems inevitable.

In particular, we update, iteratively, (i) $\pi$, $\Theta$ and $B$, and (ii) $G$, by taking into account the strengths of the available algorithms. The hybrid algorithm thus consists of two individual algorithms that carry out the respective updates:

(i) The expectation-maximisation (EM) algorithm for updating $\pi$, $\Theta$ and $B$ (Dempster et al., 1977);

(ii) The constrained-Newton method (CNM) for updating $G$ nonparametrically (Wang, 2007).

We describe, in Sections 3.3.2 and 3.3.3, these two algorithms for solving the individual likelihood maximization problems, in particular their relevant versions to the problem to be solved here, and, in Section 3.3.4, how to combine them into an efficient hybrid algorithm.

3.3.2 Updating $\pi$, $\Theta$ and $B$

The EM algorithm is the most popular tool for likelihood maximization estimation when fitting a finite mixture model (McLachlan and Krishnan, 1997; McLachlan and Peel, 2000). It has an easy implementation and is numerically stable. Its main disadvantage is slow convergence of a linear, and sometimes sublinear, order.
For a homoscedastic finite mixture, the log-likelihood function is given by

\[
l(\pi, \Theta, H) = \sum_{i=1}^{n} \log \left\{ f(x_i; \pi, \Theta, H) \right\}.
\]

To update \((\pi, \Theta, H)\), the EM iteration formulae are given by

\[
\begin{align*}
\pi'_j &= \frac{1}{n} \sum_{i=1}^{n} p_{ij}, \\
\theta'_j &= \frac{\sum_{i=1}^{n} p_{ij} x_i}{\sum_{i=1}^{n} p_{ij}}, \\
H' &= \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} (x_i - \theta_j) (x_i - \theta_j)^\top
\end{align*}
\] (3.6)

where

\[
p_{ij} = \frac{\pi_j f_{ij}}{\sum_{l=1}^{m} \pi_l f_{il}},
\]

and \(f_{ij} = f(x_i; \theta_j, H)\). That is, \((\pi, \Theta, H)\) is updated by one iteration of the EM algorithm to \((\pi', \Theta', H')\), or equivalently, \((\pi_t, \Theta_t, H_t)\) is updated to \((\pi_{t+1}, \Theta_{t+1}, H_{t+1})\), where \(t\) indicates the current iteration number.

If one restricts \(H = h^2 B\) with \(|B| = 1\), as described in Section 3.2.2, and wishes to maximize the log-likelihood function

\[
l_h(\pi, \Theta, B) = \sum_{i=1}^{n} \log \left\{ f(x_i; \pi, \Theta, h^2 B) \right\},
\]

then it is also convenient to use the EM algorithm. Despite the restriction \(|B| = 1\), the EM iterate for estimating \(B\) is simply the usual EM iterate for estimating \(H\) followed by a normalization (Celeux and Govaert, 1995):

\[
B' = H'/|H'|^{\frac{1}{2}}.
\]

(3.8)
If $B$ is further restricted to be diagonal, then $H'$ in (3.6) is diagonal, too, using only the diagonal elements of the matrix given by the right-hand side of (3.6). In other words, the $k$th diagonal element of $H'$ is given by

$$H'_{kk} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} (x_{ik} - \theta_{jk})^2,$$

where $x_{ik}$ is the $k$th element of $x_i$ and $\theta_{jk}$ the $k$th element of $\theta_j$.

The EM algorithm can be stopped by thresholding on the increase of the log-likelihood value from iteration to iteration.

### 3.3.3 Updating $G$

To estimate the NPMLE $\hat{G}$ in the multivariate case with $H$ fixed as introduced in Section 3.2.1, let us consider extending the CNM. A direct application, however, can be very costly or even infeasible in the multivariate case, since finding a local maximum of the gradient function is a multi-dimensional optimization problem and is by itself computationally not cheap, not to mention that the CNM requires this be done a large number of times. To avoid this, our proposal instead combines an algorithm for updating $\pi$ and $\Theta$ of a finite mixture, such as the EM algorithm given in Section 3.3.2, and one iteration of a modified CNM for expanding, if necessary, the support set. It is similar in spirit to the CNM-MS algorithm for fitting a semiparametric mixture (Wang, 2010).

We know from Chapter 2.3 that $\hat{G}$ is the NPMLE, if and only if $\sup_{\theta} \{ d_H(\theta; \hat{G}) \} = 0$. For an iterate $G_t$, $d_H(\theta; G_t)$ indicates where and how much its discrepancy is from $\hat{G}$ and hence where new support points are needed. However, finding all the local maxima of $d_H(\theta; G_t)$ exactly, as done by the CNM in the univariate case, can be very costly in the multivariate case, but this can be avoided in our proposed algorithm since the optimization step for fitting a finite mixture will eventually set $d_H(\theta; \cdot) = 0$ at every support point. We therefore only evaluate $d_H(\theta; G_t)$ over a sufficiently fine
grid and then decide which grid points should better be included in the support set of $G_{t+1}$.

Using a grid for gradient evaluation is to help locate roughly the local maxima of the gradient function, where their exact locations can be determined later by, say, the EM algorithm. Using a deterministic grid is possible but can be expensive to be effective in a high-dimensional space. Instead, we use a random grid, by making use of a special property of the gradient function. Note that the first term of (3.3) in fact corresponds to an $n$-component mixture density

$$d_i^*(\theta; G_t) = C^{-1} \sum_{i=1}^{n} w_i f(x_i; \theta, H),$$

(3.9)

where $w_i = f_H(x_i; G_t)^{-1}$ and $C = \sum_{i=1}^{n} w_i \int_{\Omega} f(x_i; \theta, H) d\theta$. Note that here $f(x_i; \theta, H)$, needing normalization in general, is a density function of $\theta$, not $x_i$, and, in the normal distribution case, is the normal density with mean $x_i$. Sampling from the mixture, that gives the random grid, is straightforward. This random grid is generated conditionally on $G_t$ and serves our purpose quite well, since it tends to provide more evaluation points in areas with larger gradients and ignore practically areas where the gradients are small (i.e., negatively large). The sample size, $n_g$, controls the density of the grid, and thus, by increasing $n_g$, the local maxima of a gradient function can all be found, and increasingly precisely, which ensures finding $\hat{G}$ exactly. Even if $G_t \not\rightarrow \hat{G}$, only possibly for a finite $n_g$, we note that $\sup_{\theta} \{d_H(\theta; G_t)\}$, always an upper bound of $l_H(\hat{G}) - l_H(G_t)$, is bounded above (with probability one) by a function of $n_g$, because the curvature of the gradient function is bounded with a give $H$. From our numerical experience, the discrepancy, if any, between $G_\infty$ and $\hat{G}$ is indiscernible, even using a moderate value of $n_g$, say, 1000.

To select for new support points, we adopt the support set expansion strategy of Wang (2008). Proposed in the context of survival estimation which has a discrete support space, it aims to give a controlled expansion of the support set. If too many
new support points are added at a time, it increases unnecessarily the dimensions of the quadratic programming subproblem. On the other hand, if the support set is expanded too slowly, the algorithm will need many iterations to find the full support set. Both may drastically increase computation cost. To adopt the strategy here, consider an estimate $G$ with $m$ support points. To expand its support set, choose an arbitrary coordinate, say, the $k$th, where the choice of $k$ does not make much a difference. The break points usually have the number of $m + 2$, including the current support points, the grid point with the minimum value on the the $k$th coordinate axis and the grid point with the maximum value on the $k$th coordinate axis. Then divide the $k$th coordinate axis into $m + 1$ disjoint intervals, by using two consecutive break points of the $k$th coordinates, ordered by their $k$th coordinates, and group all grid points by the intervals. All the current support points are also included in support point sets. Finally, find the grid point with the largest gradient value in each group and, if the gradient is positive, add the point to the support set. The support set can thus potentially, but only if necessary, double its size at each iteration. The strategy often quickly produces a support set of about the same size as that of the NPMLE.

With a support set given by $\Theta = (\theta_1, \ldots, \theta_m)$, one needs to update $\pi$, the mixing proportion vector. This can be readily carried out by the algorithm described in Wang (2007), or its modified version given by Wang (2010). In the multivariate case here with log-likelihood (3.4) or (3.7), first we have

$$\frac{\partial l_H}{\partial \pi} = S^T 1,$$

$$\frac{\partial^2 l_H}{\partial \pi \partial \pi^T} = -S^T S,$$

where $s_i(\pi, \theta) = \frac{\partial \log \{f(x_i; \pi, \theta, B)\}}{\partial \pi}$, $i = 1, \ldots, n$, $S = (s_1(\pi, \theta), \ldots, s_n(\pi, \theta))^T$ and
1 = (1, \ldots, 1)^T. The Taylor series expansion about \( \pi \) gives

\[
\begin{align*}
    l_H(\pi, \Theta) - l_H(\pi', \Theta) & \approx -1^T S(\pi' - \pi) + \frac{1}{2}(\pi' - \pi)S^T S(\pi' - \pi) \\
    & = \frac{1}{2} \| S\pi' - 2 \|^2 - \frac{n}{2},
\end{align*}
\]

where \( 2 = (2, \ldots, 2)^T \) and \( \| \cdot \| \) denotes the \( L_2 \)-norm. Therefore, to maximize \( l_H \) over \( \pi \) with \( \Theta \) fixed, one can iteratively solve the following least squares linear regression problem over a probability simplex:

\[
\min_{\pi'} \| S\pi' - 2 \|^2, \quad \text{subject to } \pi'^T 1 = 1, \pi' \geq 0. \tag{3.10}
\]

The unity constraint can be removed by a transformation suggested by Dax (1990), which works as follows. If \( \tilde{\pi} \) solves

\[
\min_{\pi} \| P\tilde{\pi} \|^2 + |\tilde{\pi}^T 1 - 2|^2, \quad \text{subject to } \tilde{\pi} \geq 0, \tag{3.11}
\]

where \( P \equiv (s^{(1)} - 2, \ldots, s^{(m)} - 2) \), \( s^{(j)} \) being the \( j \)th column of \( S \), then \( \tilde{\pi} / \tilde{\pi}^T 1 \) solves problem (3.10). The transformed problem (3.11) is easily solvable by the non-negative least squares (NNLS) algorithm of Lawson and Hanson (1974, p.161).

With \( H \) fixed, we adopt the computational strategy that alternates between updating the parameters \( \pi \) and \( \Theta \) by the EM and updating the nonparametric mixing distribution \( G \) by the CNM. Since one does not have to run one iteration of an algorithm for each updating, we tested a few combinations and found that running 5 iterations of the EM and 1 iteration of the CNM seemed to converge fastest. This finding happens to be consistent with what other researchers have experienced with the performance of the EM algorithm. For example, Redner and Walker (1984) reported that 95% of the change in log-likelihood value happened in the first 5 EM
3.3. COMPUTATION

iterations; see also Aitkin and Aitkin (1996). Note that we should better ensure that the final EM updating is optimal, meaning that we should only stop the EM algorithm when the increase in log-likelihood becomes negligible (and hence need to reset the maximum number of iterations to $\infty$).

In addition, since the EM algorithm may converge very slowly when there exist similar support points, which most likely will converge to one another eventually, we add an extra step that collapses similar support points. Of all potential pairs of current support points, we choose the pair that has the shortest Mahalanobis distance. Their merging by weighted averaging of the support points is accepted, if the decrease, if any, in log-likelihood after merging is negligible. The threshold value we use for accepting a merging is $10^{-4}$. This process proceeds until a pair is not accepted for merging.

We summarize the above developments in the following algorithm for computing the NPMLE in the multivariate case with $H$ fixed. We call it the CNMM algorithm which stands for the CNM method modified for multivariate observations.

**Algorithm 3 (CNMM)** Set $t = 0$ and $t_{\text{EM}} = 5$. From an initial estimate $G_0$ with finite support and $l_H(G_0) > -\infty$, repeat the following steps.

**Step1:** Run $t_{\text{EM}}$ iterations of the EM algorithm, which updates $(\Theta_t, \pi_t)$ to $(\Theta_{t+\frac{1}{2}}, \pi_{t+\frac{1}{2}})$, or equivalently $G_{t+\frac{1}{2}}$.

**Step2:** Generate a random sample $\theta^*_1, \ldots, \theta^*_n$ from density $d_H^*(\theta; G_{t+\frac{1}{2}})$ as defined by (3.9), and divide them into $m + 1$ groups, based on the $m$ intervals on an arbitrarily chosen coordinate which are defined by the current support points.

**Step3:** Let $\Theta^+$ contain $\{\theta^*_l : d_H^*(\theta^*_l; G_{t+\frac{1}{2}}) > 0, l = 1, \ldots, m\}$. Set $\Theta^+_{t+\frac{1}{2}} = (\Theta_{t+\frac{1}{2}}, \Theta^+)$ and $\pi^+_{t+\frac{1}{2}} = (\pi_{t+\frac{1}{2}}^T, 0^T)^T$. Find $\pi^-_{t+1}$, by solving problem (3.10), followed with a line search.

**Step4:** Discard all support points with zero entries in $\pi^-_{t+1}$, which gives $G_{t+1}$ with $\Theta_{t+1}$ and $\pi_{t+1}$. 
CHAPTER 3. NONPARAMETRIC MULTIVARIATE DENSITY ESTIMATION USING MIXTURES

Step5: If \( l_H(G_{t+1}) - l_H(G_t) \leq \tau \) (a small positive threshold value) and \( t_{EM} = \infty \), break. If \( l_H(G_{t+1}) - l_H(G_t) \leq \tau \), set \( t_{EM} = \infty \); otherwise, set \( t_{EM} = 5 \). Set \( t = t + 1 \).

3.3.4 The hybrid algorithm

Maximizing log-likelihood (3.4) with respect to both \( G \) and \( B \) is similar to that studied in Section 3.3.3. The algorithm proposed below is called the CNMMB, which is merely a slight extension of CNMM, by including the updating of \( B \) in the EM step, which hence maximizes, over \((\pi, \Theta, B)\), the log-likelihood function

\[
l_h(\pi, \Theta, B) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} \pi_j f(x; \theta_j, h^2 B) \right\}.
\] (3.12)

Note that (3.12) is about a finite mixture estimation and that the estimate may turn out to have fewer than \( m \) components. In fact, the EM may only produce a local maximizer of (3.12), which, however, is not really a problem here, since, if necessary, the support set will be expanded afterward. Finally, we have the following algorithm for maximizing log-likelihood (3.4).

**Algorithm 4 (CNMMB)**. Hold \( h \) at a fixed value and set \( t = 0 \) and \( t_{EM} = 5 \).

From an initial \( G_0 \) with finite support, a positive definite \( B_0 \) and \( l_h(G_0, B_0) > -\infty \), repeat the following steps.

**Step1**: Run \( t_{EM} \) iterations of the EM, which updates \((\Theta_t, \pi_t, B_t)\) to \((\Theta_{t+\frac{1}{2}}, \pi_{t+\frac{1}{2}}, B_{t+1})\).

Steps 2–5: Much the same as steps 2–5 in CNMM.

To choose initial values, one can in general use the standardized sample covariance matrix, or the diagonal matrix, for \( B_0 \). For \( G_0 \), one may pick some observed points as long as the initial mixture density gives a good coverage of all the observations. In the case of finding a series of the estimates corresponding to a grid of \( h \) as described in Section 3.2.3, one can, e.g., start with a large value of \( h \) that gives a unicomponent
mixture estimate, and then compute and use in turn the estimates \( \hat{G}_h \) and \( \hat{B}_h \) as the initial values for the subsequent \( h \)-values.

### 3.4 Demonstration of finding an NPMLE

In this section, a simulated data example is used to demonstrate the process of finding an NPMLE by using the CNMM. This data set of size 500 was generated from a bivariate normal mixture distribution with a bimodal shape. The density function of this bivariate normal mixture is parameterized by \( \frac{1}{2} \mathcal{N}(-1, 0, \frac{4}{9}, \frac{2}{9}, \frac{4}{9}) + \frac{1}{2} \mathcal{N}(1, 0, \frac{4}{9}, \frac{2}{9}, \frac{4}{9}) \), where \( \mathcal{N}(\mu_1, \mu_2, \sigma_{11}, \sigma_{12}, \sigma_{22}) \) denotes a bivariate normal with mean \((\mu_1, \mu_2)^T\) and covariance matrix \[
\begin{pmatrix}
\sigma_{11} & \sigma_{12} \\
\sigma_{12} & \sigma_{22}
\end{pmatrix}
\].

A contour plot of the density function for this distribution is shown in Fig. 3.1.

![A bimodal distribution.](image)

**Figure 3.1:** A bimodal distribution.

To demonstrate the process of the CNMM, we need to fix the \( H \) of the density estimate beforehand. Firstly, the AIC was used to select an optimal \( h \)-value from a grid of 10 potential values, which were evenly distributed from \( \sqrt{0.1}s \) to \( s \). Here \( d \) is the dimensionality and \( s \) the volume parameter value of the sample covariance matrix. The CNMMB then was used to find a density estimate for each \( h \)-value. A random grid of 200 points were used for this example. For this data set, the optimal
h-value was 0.53 and

$$\hat{H} = \begin{pmatrix} 0.2803 & 0.1166 \\ 0.1166 & 0.3244 \end{pmatrix}.$$  

With the given $\hat{H}$, the CNMM used 12 iterations to find the NPMLE. Fig. 3.2 shows the results of three selected iterations of the CNMM in the process of finding the NPMLE $\hat{G}$. The three rows in Fig. 3.2 represent the results of the 1st, 4th and 12th iterations after step 2 of the CNMM. Each of them contains two subplots.

The algorithm began with one component normal mixture with mean the sample mean and ended with a mixture of 9 components. We can see that there are more grid points generating around local maxima and less grid points generated far away from local maxima. This gives a high probability to find a new support point that is very close to a local maximum. It can be seen that after the last iteration the newly found support points are exactly the same as the current support points, which means there is no other support point which has a larger gradient value and may potentially help increase the log-likelihood value. The contour plot of the density estimate of the 12th iteration is very similar to the contour plot of the true density function shown in Fig. 3.1.

### 3.5 Concluding Remarks

In this chapter, we study nonparametric multivariate density estimation using mixture distributions. In particular, we use nonparametric and semiparametric mixture distributions for this purpose. Our new procedure consider the likelihood function of all unknown parameters except the volume of the component covariance matrix, which is to be held fixed at each value of a pre-given grid for each likelihood maximization and to be selected via a model selection criterion, such as the AIC. This
resolves the dilemma that maximizing the likelihood with respect to the whole component covariance matrix always gives singular and practically useless solutions and that determining the whole component covariance matrix via a model selection criterion is computationally infeasible, even in a moderate-dimensional space. Efficient computation methods are also described.

Although we did not study the issue of variable selection, it should be fairly straightforward for the new procedure to deal with it. For example, one can start with a density estimate with all variables included and then progressively remove the one that results in the least reduction in the likelihood. Since all estimation is likelihood-based, the AIC can be used directly to determine the final density estimate with an appropriate set of variables. It looks much more difficult for the KDE approach to deal with variable selection.

A somewhat related method has been proposed by Jones and Henderson (2009), who extended the KDE with a fixed bandwidth by using the maximum likelihood for finding better kernel centers than the observed points, while keeping the weights remain $1/n$. We note that the resulting estimate lies between the KDE and NPMLE, for the latter would be obtained if no restriction is enforced on the weights. Jones and Henderson only studied their method in the univariate case, but it may potentially be applicable to multivariate observations.
Figure 3.2: 3 iterations of the CNMM for finding the NPMLE $\hat{G}$. Each left panel shows contour lines of the gradient function with random grids in grey and “plus” and current support points in red and “star” and the newly found support points are in red and “circle”. Each right panel shows contour lines of the corresponding density estimate with observations in blue and “plus” and current component centers in red and “star”.
Chapter 4

Numerical Studies for Nonparametric Multivariate Density Estimation

4.1 Simulation Studies

4.1.1 Setup

We carried out simulation studies in R (R Development Core Team, 2012) to compare the performance of the MDE and KDE in the multivariate case. For the KDE, four bandwidth matrix selection methods were studied: the product kernel estimator (PD) (Scott, 1992), the adaptive kernel estimator (AD) (Silverman, 1986), the plug-in selector (PI) of Duong and Hazelton (2003) and the smooth cross-validation selector (SCV) of Duong and Hazelton (2005). The PD and AD estimators with diagonal matrices are available in the R package pdfCluster (Azzalini et al., 2012; Menardi and Azzalini, 2013), while the PI and SCV selectors with full matrices are available in the R package ks (Duong, 2008). Duong (2008) has shown through empirical studies that the KDE’s based on diagonal bandwidth matrices do not perform as well as their counterparts based on full bandwidth matrices, and also that the PI and SCV selectors perform better than some other selectors with full bandwidth matrices, such as the least squares cross-validation selector (Rudemo, 1982;
Bowman, 1984) and the biased cross-validation selector (Sain et al., 1994). For the MDE, we only report the results based on full bandwidth matrices. The estimates which restrict $B$ to be an identity or diagonal matrix were also studied, but their performance was, as expected, worse and is thus not reported here.

Eighteen bivariate mixtures were considered and they contain a number of interesting characteristics, including correlation, skewness, homoscedasticity, heteroscedasticity, multi-modality and heavy-tailness. In particular, homogenous normal mixtures, heterogenous normal mixtures and $t$-distribution mixtures are considered in the simulation studies. A multivariate $t$ density function is given by

$$t_v(\mu, \Sigma) = \frac{\Gamma[(v+d)/2]}{\Gamma(v/2)d^{d/2}2^{d/2}|\Sigma|^{1/2}} \left[1 + \frac{1}{v}(x - \mu)^T \Sigma^{-1} (x - \mu)(x+d)/2\right]^{-(v+d)/2},$$

where $x \in \mathbb{R}^d$, $\mu \in \Omega \subset \mathbb{R}^d$, $\mu$ is the location parameter, $\Sigma$ the covariance matrix and $v$ the degrees of freedom, and $\Gamma()$ is the Gamma function. Among the 18 mixtures, six are normal mixtures, six are $t$ mixtures with 5 degrees of freedom and six are $t$ mixtures with 10 degrees of freedom, as listed in Table 4.1. For each normal mixture, there are correspondingly $t$ mixtures that have the same location parameters and covariance matrices as the normal mixture, 5 and 10 degrees of freedom, respectively.

As described in Section 3.2.3, we used the AIC to select an appropriate $h$-value, from a grid of 10 potential values that were evenly distributed from $d^{\sqrt{0.1}s}$ to $s$. Here $d$ is the dimensionality and $s$ the volume parameter value of the sample covariance matrix, which reduces to the sample standard deviation in the univariate case. Starting from $h = s$, a density estimate was found by the CNMNB for each $h$-value.

To assess the performance of a density estimator, we used three loss functions, namely, the integrated square error (ISE), the Kullback-Leibler (KL) divergence and
### 4.1. SIMULATION STUDIES

<table>
<thead>
<tr>
<th>Name</th>
<th>Type of Mixture</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A1) Correlated</td>
<td>Single Normal</td>
<td>$N(0, 0, 1, \frac{7}{10}, 1)$</td>
</tr>
<tr>
<td>(A2) Correlated</td>
<td>Single $t$</td>
<td>Same means and covariance matrices as (A1) with 5 degrees of freedom</td>
</tr>
<tr>
<td>(A3) Correlated</td>
<td>Single $t$</td>
<td>Same means and covariance matrices as (A1) with 10 degrees of freedom</td>
</tr>
<tr>
<td>(B1) Skewed</td>
<td>Normal mixture</td>
<td>$\sum_{j=0}^{7} \frac{1}{8} N(1 - (\frac{4}{5})^j (3, -3), (\frac{4}{5})^2 (1, -\frac{9}{10}, 1))$</td>
</tr>
<tr>
<td>(B2) Skewed</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (B1) with 5 degrees of freedom</td>
</tr>
<tr>
<td>(B3) Skewed</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (B1) with 10 degrees of freedom</td>
</tr>
<tr>
<td>(C1) Bimodal 1</td>
<td>Normal mixture</td>
<td>$\frac{1}{2} N(-1, 0, \frac{4}{7}, \frac{2}{7}, \frac{4}{7}) + \frac{1}{2} N(1, 0, \frac{4}{7}, \frac{2}{7}, \frac{4}{7})$</td>
</tr>
<tr>
<td>(C2) Bimodal 1</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (C1) with 5 degrees of freedom</td>
</tr>
<tr>
<td>(C3) Bimodal 1</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (C1) with 10 degrees of freedom</td>
</tr>
<tr>
<td>(D1) Bimodal 2</td>
<td>Normal mixture</td>
<td>$\frac{1}{2} N(-1, 1, \frac{4}{7}, \frac{1}{7}, \frac{4}{7}) + \frac{1}{2} N(0, 0, \frac{4}{7}, 0, \frac{4}{7})$</td>
</tr>
<tr>
<td>(D2) Bimodal 2</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (D1) with 5 degrees of freedom</td>
</tr>
<tr>
<td>(D3) Bimodal 2</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (D1) with 10 degrees of freedom</td>
</tr>
<tr>
<td>(E1) Bimodal 3</td>
<td>$t$ mixture</td>
<td>$\frac{1}{2} N(0.3, 0.2, 0.7, 0.3, 0.2) + \frac{1}{2} N(1, 1.2, 0.7, -0.3, 0.2)$</td>
</tr>
<tr>
<td>(E2) Bimodal 3</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (E1) with 5 degrees of freedom</td>
</tr>
<tr>
<td>(E3) Bimodal 3</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (E1) with 10 degrees of freedom</td>
</tr>
<tr>
<td>(F1) Trimodal</td>
<td>$t$ mixture</td>
<td>$\frac{3}{7} N(-1, 0, 9, \frac{63}{10}, \frac{49}{10}) + \frac{3}{7} N(1, \frac{2}{\sqrt{3}}, 9, 0, \frac{49}{10}) + \frac{1}{7} N(1, -\frac{2}{\sqrt{3}}, 9, 0, \frac{49}{10})$</td>
</tr>
<tr>
<td>(F2) Trimodal</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (F1) with 5 degrees of freedom</td>
</tr>
<tr>
<td>(F3) Trimodal</td>
<td>$t$ mixture</td>
<td>Same means and covariance matrices as (F1) with 10 degrees of freedom</td>
</tr>
</tbody>
</table>

Table 4.1: 18 bivariate mixture distributions used in the simulation studies, where $N(\mu_1, \mu_2, \sigma_{11}, \sigma_{12}, \sigma_{22})$ denotes a bivariate normal with mean $(\mu_1, \mu_2)^T$ and covariance matrix $\begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix}$.  


the Hellinger distance (HD), given by, respectively,

\[
\text{ISE}(f, \hat{f}) = \int \{\hat{f}(x) - f(x)\}^2 dx,
\]

\[
\text{KL}(f, \hat{f}) = \int f(x) \log \left\{ \frac{f(x)}{\hat{f}(x)} \right\} dx,
\]

\[
\text{HD}(f, \hat{f}) = \int \left\{ \hat{f}(x)^{1/2} - f(x)^{1/2} \right\}^2 dx,
\]

where \( \hat{f} \) is an estimate of the true density \( f \). The expectation of a loss function with respect to \( f \) can then be used to assess the performance of an estimator. To calculate the loss of an estimate \( \hat{f} \), which is an integral, we use Monte Carlo integration, specifically the importance sampling technique. This is achieved by transforming each integrand to one divided by the true density \( f \) and by computing the sample mean of the transformed integrand evaluated with a large random sample (of size 10000) generated from \( f \).

### 4.1.2 Normal mixtures

The contour plots of density functions for the six normal mixtures are shown in Fig. 4.1 and parameters for each normal mixture are listed in Table 4.1. For each distribution, we generated 100 data sets of size 100 and 100 data sets of size 500.

Table 4.2 provides a summary of the simulation results for six normal mixtures. Each entry in the table is one of the three expected losses, obtained by averaging over 100 repeated samplings, with standard error given in parentheses. Each value in boldface is the smallest expected loss among the five estimators in each case. It is clear from Table 4.2 that the MDE outperforms, often outstandingly, the four KDE’s for all six distributions, in terms of all three performances measures. This outperformance is more significant for \( n = 500 \), giving a between 65% and 15% reduction in mean loss.

It can also be seen that the performance of an estimator differs in the loss function
4.1. SIMULATION STUDIES

![Contour plots for the densities of the 6 distributions of normal mixtures given in Table 4.1.](image)

Figure 4.1: Contour plots for the densities of the 6 distributions of normal mixtures given in Table 4.1.

used and the underlying distribution. For example, the MDE outperforms the KDE more obviously in terms of the mean KL than in terms of the mean ISE. This is likely due to their specific bandwidth selection methods that target different performance measures. Moreover, the MDE performs much better, understandably, for homoscedastic mixture distributions, A1 and C1, than for heteroscedastic mixture distributions, D1, E1 and F1, and the strongly skewed distribution B1.

It is also interesting to see from the results that the MDE may sometimes have a larger standard error, or a higher variance, than the KDE. However, its smaller bias, thanks to its deconvolution nature, outweighs this shortcoming, which results in a reduction in the overall mean errors.

For distributions A1–F1, the MDE has most often 2, 3, 2, 3, 7 and 3 mixture components, respectively, with the 100 simulated data sets, for both $n = 100$ and
CHAPTER 4. NUMERICAL STUDIES FOR NONPARAMETRIC MULTIVARIATE DENSITY ESTIMATION

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(A1)</th>
<th>(B1)</th>
<th>(C1)</th>
<th>(D1)</th>
<th>(E1)</th>
<th>(F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KDE-PD</td>
<td>38.80 (2.61)</td>
<td>95.39 (2.59)</td>
<td>13.18 (1.19)</td>
<td>13.27 (1.32)</td>
<td>35.58 (1.59)</td>
<td>11.86 (0.84)</td>
</tr>
<tr>
<td>KDE-AD</td>
<td>29.99 (2.23)</td>
<td>79.57 (2.39)</td>
<td>13.67 (2.13)</td>
<td>15.10 (1.47)</td>
<td>33.13 (1.74)</td>
<td>10.63 (0.72)</td>
</tr>
<tr>
<td>KDE-PI</td>
<td>18.84 (1.41)</td>
<td>34.73 (1.79)</td>
<td>9.62 (0.87)</td>
<td>11.95 (1.13)</td>
<td>24.62 (1.60)</td>
<td>10.03 (0.59)</td>
</tr>
<tr>
<td>KDE-SCV</td>
<td>13.43 (1.36)</td>
<td>23.05 (1.78)</td>
<td>10.09 (0.99)</td>
<td>12.51 (0.89)</td>
<td>26.25 (1.89)</td>
<td>11.09 (0.60)</td>
</tr>
<tr>
<td>MDE</td>
<td>6.58 (1.50)</td>
<td>21.64 (11.10)</td>
<td>5.97 (0.90)</td>
<td>10.89 (2.20)</td>
<td>23.51 (5.24)</td>
<td>7.84 (0.91)</td>
</tr>
</tbody>
</table>

Mean Integrated Square Error (×10³)

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(A1)</th>
<th>(B1)</th>
<th>(C1)</th>
<th>(D1)</th>
<th>(E1)</th>
<th>(F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KDE-PD</td>
<td>160.61 (9.51)</td>
<td>219.89 (4.77)</td>
<td>64.23 (4.42)</td>
<td>53.09 (4.85)</td>
<td>133.55 (6.84)</td>
<td>79.19 (6.80)</td>
</tr>
<tr>
<td>KDE-AD</td>
<td>140.41 (7.15)</td>
<td>203.17 (5.01)</td>
<td>61.14 (5.29)</td>
<td>53.10 (3.58)</td>
<td>127.92 (4.31)</td>
<td>91.73 (6.48)</td>
</tr>
<tr>
<td>KDE-PI</td>
<td>69.56 (3.84)</td>
<td>98.30 (4.28)</td>
<td>52.67 (2.91)</td>
<td>53.47 (3.20)</td>
<td>97.34 (10.60)</td>
<td>59.32 (3.70)</td>
</tr>
<tr>
<td>KDE-SCV</td>
<td>48.13 (4.41)</td>
<td>60.20 (4.47)</td>
<td>57.38 (4.34)</td>
<td>61.46 (4.25)</td>
<td>107.86 (12.66)</td>
<td>66.45 (4.23)</td>
</tr>
<tr>
<td>MDE</td>
<td>44.58 (11.04)</td>
<td>83.90 (6.49)</td>
<td>54.47 (6.62)</td>
<td>79.62 (12.30)</td>
<td>144.20 (6.39)</td>
<td>90.31 (11.81)</td>
</tr>
</tbody>
</table>

Mean Kullback-Leibler Divergence (×10³)

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(A1)</th>
<th>(B1)</th>
<th>(C1)</th>
<th>(D1)</th>
<th>(E1)</th>
<th>(F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>n = 500</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KDE-PD</td>
<td>141.17 (2.82)</td>
<td>311.03 (3.75)</td>
<td>50.57 (1.97)</td>
<td>47.79 (3.32)</td>
<td>101.78 (0.32)</td>
<td>66.84 (0.45)</td>
</tr>
<tr>
<td>KDE-AD</td>
<td>144.85 (3.08)</td>
<td>297.84 (1.83)</td>
<td>57.74 (2.13)</td>
<td>57.36 (3.77)</td>
<td>109.40 (0.28)</td>
<td>67.18 (0.33)</td>
</tr>
<tr>
<td>KDE-PI</td>
<td>163.56 (8.88)</td>
<td>103.91 (6.38)</td>
<td>40.31 (2.75)</td>
<td>38.22 (1.45)</td>
<td>61.39 (3.36)</td>
<td>44.69 (2.10)</td>
</tr>
<tr>
<td>KDE-SCV</td>
<td>77.54 (5.62)</td>
<td>51.74 (3.44)</td>
<td>39.06 (2.78)</td>
<td>38.59 (1.29)</td>
<td>61.57 (3.34)</td>
<td>44.51 (2.66)</td>
</tr>
<tr>
<td>MDE</td>
<td>45.43 (17.27)</td>
<td>25.56 (1.89)</td>
<td>18.91 (1.72)</td>
<td>26.88 (3.28)</td>
<td>43.29 (2.49)</td>
<td>30.89 (3.13)</td>
</tr>
</tbody>
</table>

Mean Hellinger Distance (×10³)

To investigate the AIC criterion on selecting an optimal value of \( h \), we carried out an experiment. We generated a training data set of size 500 and a test data set of size 10000 based on the bimodal distribution C1. A grid of 10 volume values were used to estimate density functions for this training data set by the MDE, respectively. For each density estimate, the AIC, ISE, KL and HD are calculated.

The AIC values, and the errors of the ISE, KL and HD vs a grid of 10 \( h \)-values are shown in Fig. 4.2. Here different colors represent different path, while the left y-axis is for the ISE, KL and HD, and the right y-axis is for the AIC. It can be

Table 4.2: Simulation results for 6 target distributions of normal mixtures.

\( n = 500 \). This gives a substantial reduction by the MDE in terms of the space needed to store a model, as compared with the KDE that has to store all observations.

To investigate the AIC criterion on selecting an optimal value of \( h \), we carried out an experiment. We generated a training data set of size 500 and a test data set of size 10000 based on the bimodal distribution C1. A grid of 10 volume values were used to estimate density functions for this training data set by the MDE, respectively. For each density estimate, the AIC, ISE, KL and HD are calculated.

The AIC values, and the errors of the ISE, KL and HD vs a grid of 10 \( h \)-values are shown in Fig. 4.2. Here different colors represent different path, while the left y-axis is for the ISE, KL and HD, and the right y-axis is for the AIC. It can be
4.1. SIMULATION STUDIES

57

seen that the AIC reaches the smallest at the point of $h = 0.587$, where all the loss functions attain their smallest values. This suggest that the AIC works very well in this case.

4.1.3 Mixtures of $t$-distributions

The contour plots of density functions for the twelve mixtures of $t$-density with 5 degrees of freedom and 10 degrees of freedom are shown in Fig. 4.3 and Fig. 4.4, respectively. The heavy-tailness can be seen by the increasing gaps between different contour lines when moving away from the modes. For each distribution, we generated 100 data sets of size 500.

A summary of the simulation results for $t$ mixtures is given in Table 4.3. All of the five estimators perform better for $t$-distributions with 10 degrees of freedom than for these with 5 degrees of freedom, which indicates that both the KDEs and the MDE favor distributions with normal tails.

The results also show that the MDE gives the smallest, or nearly the smallest,
errors, regardless of performance measures. Therefore, we conclude that the MDE very much dominates the KDEs on densities with heavy tails.

For distributions A2–F2 where \(n = 500\) and the \(t\) distributions have 5 degrees of freedom, the MDE mostly contained 3, 5, 4, 7 and 5 mixture components, respectively. While for distribution A3–F3, the MDE often contained 2, 4, 3, 7 and 3 mixture components, respectively. It is because tails are heavier when the degrees of freedom is 5 and hence the MDE needs to use more components.

Figure 4.3: Contour plots for the 6 \(t\) mixtures with 5 degrees of freedom as given in Table 4.1.
4.2 Real Data Sets

4.2.1 Setup

In this section, we study two real-world problems that arise from medical and food science, respectively. The two data sets, downloaded from the Machine Learning Repository at the University of California at Irvine, are the Indian Liver Patient data and the Wine data. They are studied in Sections 4.2.2 and 4.2.3, respectively. All variables in both of the original data sets have been log-transformed.

Since true underlying densities for real-world data sets are unknown, we used two comparisons to assess performance to provide a stern test for the MDE. The first comparison used 10-fold cross-validation, with 100 repetitions. Two loss functions,
the KDE with the full bandwidth matrices that was found by the SCV. 100 data from the above formulae when losses are computed.

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(A2)</th>
<th>(B2)</th>
<th>(C2)</th>
<th>(D2)</th>
<th>(E2)</th>
<th>(F2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDE-PD</td>
<td>28.13 (1.13)</td>
<td>68.27 (1.81)</td>
<td>6.44 (0.39)</td>
<td>7.15 (0.40)</td>
<td>21.52 (1.17)</td>
<td>6.70 (0.28)</td>
</tr>
<tr>
<td>KDE-AD</td>
<td>15.17 (0.81)</td>
<td>47.01 (1.58)</td>
<td>3.42 (0.40)</td>
<td>3.88 (0.30)</td>
<td>12.22 (0.78)</td>
<td>3.17 (0.24)</td>
</tr>
<tr>
<td>KDE-PI</td>
<td>5.10 (0.48)</td>
<td>11.75 (0.70)</td>
<td>3.93 (0.40)</td>
<td>4.98 (0.32)</td>
<td>10.82 (0.55)</td>
<td>3.64 (0.26)</td>
</tr>
<tr>
<td>KED-SCV</td>
<td>4.10 (0.39)</td>
<td>8.82 (0.80)</td>
<td>4.36 (0.43)</td>
<td>5.12 (0.34)</td>
<td>10.93 (0.52)</td>
<td>3.93 (0.29)</td>
</tr>
<tr>
<td>MDE</td>
<td>2.90 (0.61)</td>
<td>8.53 (0.98)</td>
<td>3.14 (0.42)</td>
<td>3.44 (0.38)</td>
<td>9.42 (0.52)</td>
<td>2.89 (0.35)</td>
</tr>
</tbody>
</table>

| Mean Integrated Square Error ($\times 10^4$) |
|---------------------------------|------|------|------|------|------|------|
| KDE-PD                          | 270.12 (8.57) | 503.98 (11.10) | 161.23 (11.30) | 173.94 (9.68) | 179.53 (5.74) | 126.49 (5.91) |
| KDE-AD                          | 249.48 (6.46) | 487.54 (14.25) | 69.12 (2.67) | 91.27 (7.81) | 155.71 (8.42) | 46.06 (2.43) |
| KDE-PI                          | 154.22 (6.70) | 272.11 (28.50) | 219.87 (13.49) | 234.04 (10.65) | 254.26 (10.84) | 200.01 (9.32) |
| KED-SCV                         | 108.41 (6.12) | 120.82 (16.53) | 178.58 (9.48) | 217.81 (11.34) | 254.97 (9.53) | 174.92 (7.94) |
| MDE                             | 55.75 (5.60) | 83.90 (6.49) | 58.74 (2.46) | 89.20 (8.85) | 89.43 (4.47) | 91.00 (4.97) |

| Mean Kulback-Leibler Divergence ($\times 10^4$) |
|---------------------------------|------|------|------|------|------|------|
| KDE-PD                          | 99.41 (4.25) | 405.66 (9.17) | 32.49 (1.46) | 36.56 (2.49) | 67.65 (3.45) | 38.99 (1.96) |
| KDE-AD                          | 87.79 (4.09) | 276.79 (7.83) | 19.95 (1.77) | 30.42 (1.18) | 61.05 (3.93) | 23.74 (1.34) |
| KDE-PI                          | 46.04 (1.76) | 461.11 (7.28) | 30.80 (1.20) | 36.32 (3.40) | 47.73 (1.84) | 34.50 (1.85) |
| KED-SCV                         | 37.45 (1.70) | 338.42 (5.91) | 28.96 (1.32) | 35.30 (3.16) | 47.74 (1.83) | 33.05 (1.75) |
| MDE                             | 19.33 (1.16) | 84.67 (5.89) | 17.42 (1.49) | 27.63 (5.50) | 39.44 (1.80) | 23.73 (1.20) |
| Estimator (A3)                  | 22.86 (1.36) | 67.84 (1.14) | 3.14 (0.30) | 5.88 (0.42) | 20.18 (0.47) | 6.71 (0.39) |
| 10 degrees of freedom           | 12.29 (0.74) | 47.46 (1.16) | 3.58 (0.43) | 3.58 (0.35) | 12.91 (0.37) | 3.45 (0.37) |
| KDE-SCV                         | 5.53 (0.49) | 11.22 (0.69) | 3.49 (0.34) | 4.26 (0.42) | 11.48 (0.44) | 3.96 (0.45) |
| KDE-PD                          | 4.26 (0.47) | 7.63 (0.47) | 3.61 (0.39) | 4.37 (0.44) | 11.96 (0.47) | 4.28 (0.45) |
| Mean Integrated Square Error ($\times 10^4$) |
|---------------------------------|------|------|------|------|------|------|
| KDE-PD                          | 26.58 (4.36) | 30.65 (2.30) | 24.68 (2.31) | 35.88 (2.52) | 74.02 (4.71) | 36.53 (1.20) |
| KDE-AD                          | 80.60 (3.90) | 170.50 (2.19) | 25.92 (0.93) | 26.30 (1.73) | 59.70 (0.74) | 35.30 (1.70) |
| KDE-PI                          | 66.82 (2.54) | 141.52 (2.02) | 22.51 (1.03) | 23.48 (1.19) | 55.71 (0.58) | 28.35 (1.51) |
| KED-SCV                         | 34.31 (1.54) | 45.07 (1.44) | 23.33 (0.66) | 23.60 (1.31) | 37.61 (0.96) | 25.60 (1.55) |
| MDE                             | 11.82 (2.18) | 14.67 (1.41) | 11.36 (0.66) | 17.37 (1.70) | 29.67 (1.69) | 16.95 (0.64) |

Table 4.3: Simulation results for 12 target distributions of t-density mixtures.

The second comparison was made by fitting a density to each real data set using the KDE with the full bandwidth matrices that was found by the SCV. 100 data
sets of size 500 were then generated from the fitted density. The ISE and KL were used to compare the performance of the KDEs and MDE.

We used ISE(CV) and KL(CV) to indicate the ISE and KL obtained from 100 times 10-fold cross-validation, and ISE(KDE) and KL(KDE) the ISE and KL obtained by using the KDE fits to generate new data. For ISE(KDE) and KL(KDE), we only reported results for \( d \leq 4 \). Because for \( d > 4 \), the two KDEs with full matrices were computationally too slow to find. In our attempt to find the 6-dimensional KDE’s with the \texttt{ks} package (Duong, 2008), our computer (with sufficient size of memory) appears to have frozen and we had to terminate the process after 2 hours of running. The \texttt{ks} package simply does not allow for \( d > 6 \). Scott (1992, p.198) points out that the direct estimation of the full density of all kernel methods does not perform well beyond six dimensions.

### 4.2.2 The Indian Liver Patient data

The Indian Liver Patient data, collected from the northeast of Andhra Pradesh, India, contains a total of 583 observations (416 patients with liver disease and 167 healthy people) and 8 variables. Here we only consider estimating the distribution for the 416 patients, using \( d = 2, 4, 6 \) and 8 variables, respectively, by including variables in the following order: Total Bilirubin, Albumin, SGOT, Total Proteins, Alkphos, SGPT, Direct Bilirubin and A/G Ratio. The ordering of these variables is not pivotal for the results obtained. Fig. 4.5 shows the pairwise scatter plots for this data set.

The results of the study are given in Table 4.4. They again show that the MDE is superior to all the KDE’s in terms of the two comparison methods with two performance measures, for all \( ds \). When \( d = 2 \), the MDE consisted of 20 components, with \( h = 0.213 \) \((= 0.392 s)\). Fig. 4.6 shows the contour plots of the density estimates of the PI, SCV and MDE, superimposed on a scatterplot of the data. Their estimated
covariance/bandwidth matrices are, respectively,

\[
\hat{\Sigma}_{\text{PI}} = \begin{pmatrix}
0.0439 & 0.0008 \\
0.0008 & 0.0058
\end{pmatrix},
\]

\[
\hat{\Sigma}_{\text{SCV}} = \begin{pmatrix}
0.0517 & 0.0012 \\
0.0012 & 0.0095
\end{pmatrix},
\]

\[
\hat{\Sigma}_{\text{MDE}} = \begin{pmatrix}
0.0487 & 0.0081 \\
0.0081 & 0.0436
\end{pmatrix}.
\]
4.2. REAL DATA SETS

<table>
<thead>
<tr>
<th>Performance</th>
<th>KDE-PD</th>
<th>KDE-AD</th>
<th>KDE-PI</th>
<th>KDE-SCV</th>
<th>MDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d = 2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean ISE(CV)</td>
<td>$-0.538 (0.006)$</td>
<td>$-0.626 (0.009)$</td>
<td>$-0.659 (0.010)$</td>
<td>$-0.634 (0.009)$</td>
<td>$-0.724 (0.012)$</td>
</tr>
<tr>
<td>Mean ISE(KDE)</td>
<td>$0.035 (0.001)$</td>
<td>$0.019 (0.001)$</td>
<td>$0.016 (0.001)$</td>
<td>$0.019 (0.001)$</td>
<td>$0.014 (0.001)$</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$1.383 (0.014)$</td>
<td>$1.348 (0.017)$</td>
<td>$1.331 (0.021)$</td>
<td>$1.330 (0.019)$</td>
<td>$1.298 (0.020)$</td>
</tr>
<tr>
<td>Mean KL(KDE)</td>
<td>$0.096 (0.003)$</td>
<td>$0.095 (0.004)$</td>
<td>$0.072 (0.007)$</td>
<td>$0.069 (0.005)$</td>
<td>$0.059 (0.005)$</td>
</tr>
<tr>
<td>$d = 4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean ISE(CV)</td>
<td>$-0.484 (0.008)$</td>
<td>$-0.710 (0.026)$</td>
<td>$-0.801 (0.012)$</td>
<td>$-0.700 (0.010)$</td>
<td>$-0.866 (0.042)$</td>
</tr>
<tr>
<td>Mean ISE(KDE)</td>
<td>$0.093 (0.003)$</td>
<td>$0.054 (0.001)$</td>
<td>$0.055 (0.002)$</td>
<td>$0.055 (0.003)$</td>
<td>$0.044 (0.003)$</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$2.254 (0.033)$</td>
<td>$1.927 (0.042)$</td>
<td>$2.187 (0.050)$</td>
<td>$2.025 (0.038)$</td>
<td>$1.775 (0.090)$</td>
</tr>
<tr>
<td>Mean KL(KDE)</td>
<td>$0.403 (0.005)$</td>
<td>$0.409 (0.006)$</td>
<td>$0.367 (0.010)$</td>
<td>$0.296 (0.008)$</td>
<td>$0.268 (0.009)$</td>
</tr>
<tr>
<td>$d = 6$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean ISE(CV)</td>
<td>$-0.123 (0.005)$</td>
<td>$-0.419 (0.033)$</td>
<td>—</td>
<td>—</td>
<td>$-0.458 (0.021)$</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$3.820 (0.051)$</td>
<td>$3.636 (0.073)$</td>
<td>—</td>
<td>—</td>
<td>$3.441 (0.082)$</td>
</tr>
<tr>
<td>$d = 8$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean ISE(CV)</td>
<td>$-0.053 (0.001)$</td>
<td>$-0.392 (0.033)$</td>
<td>—</td>
<td>—</td>
<td>$-0.615 (0.032)$</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$4.807 (0.064)$</td>
<td>$4.493 (0.089)$</td>
<td>—</td>
<td>—</td>
<td>$3.286 (0.190)$</td>
</tr>
</tbody>
</table>

Table 4.4: Results of five estimators for the Indian Liver Patient data set with $d = 2, 4, 6$ and 8 variables. Standard errors are given in parentheses.

Here the covariance matrices for the KDEs are actually bandwidth matrices.

Figure 4.6: Contour plots of density estimates for the bivariate Indian Liver Patient data

When $d = 4, 6$ and 8, the MDE used $\hat{h} = 0.254 (= 0.611s), 0.379 (= 0.858s)$ and $0.324 (= 0.862s)$, and had 45, 39 and 44 mixture components, respectively. All the univariate marginal densities by the MDE for $d = 4, 6$ and 8 are shown in Fig. 4.7. As $d$ increases, it seems that the models selected by the AIC tend to be over-smoothed, despite that the AIC is generally considered conservative. A plausible reason is that the regularity conditions for the asymptotic normality of maximum likelihood estimation are not satisfied by mixture models. How to readily
select among these multivariate mixture models is worth further research. One can, of course, always resort to cross-validation to find an appropriate value for \( h \) at the price of higher computation cost.

![Figure 4.7: Univariate marginal densities by the MDE for the Indian Liver Patient data.](image)

### 4.2.3 The Wine data

The Wine data were originally collected for a chemical analysis of wines which were grown in different cultivars of the same region in Italy. The data set contains 178 observations, and 13 variables that are indicative of features of different types of wine. Here we consider the estimation of the distributions with 2, 5, 9 and 13 variables, by including variables in the same following order as given in the original data set: Alcohol, Malic acid, Ash, Alcalinity of ash, Alcohol, Magnesium and Total phylloids, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue,
OD280/OD315 of diluted wines and Prolin. Fig. 4.8 shows the pairwise scatter plots for this data set.

![Pairwise scatter plots for the Wine data.](image)

Table 4.5 gives the calculated mean losses of the five density estimators. The MDE dominates all the KDE’s regardless of comparison methods and loss functions. When $d = 2$, the MDE is an 11-component mixture with $\hat{h} = 0.0783$ ($= 0.468s$). The
### Table 4.5: Results of five estimators for the Wine data set with $d = 2, 5, 9$ and $13$ variables. Standard errors are given in parentheses.

<table>
<thead>
<tr>
<th>Performance Measure</th>
<th>KDE-PD</th>
<th>KDE-AD</th>
<th>KDE-PI</th>
<th>KDE-SCV</th>
<th>MDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean ISE(CV)</td>
<td>$-4.170$ (0.002)</td>
<td>$-4.543$ (0.003)</td>
<td>$-4.502$ (0.011)</td>
<td>$-4.400$ (0.010)</td>
<td>$-5.056$ (0.021)</td>
</tr>
<tr>
<td>Mean ISE(KDE)</td>
<td>$0.147$ (0.007)</td>
<td>$0.113$ (0.006)</td>
<td>$0.124$ (0.006)</td>
<td>$0.128$ (0.007)</td>
<td>$0.103$ (0.007)</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$-0.827$ (0.016)</td>
<td>$-0.838$ (0.019)</td>
<td>$-0.859$ (0.017)</td>
<td>$-0.853$ (0.015)</td>
<td>$-0.915$ (0.024)</td>
</tr>
<tr>
<td>Mean KL(KDE)</td>
<td>$0.049$ (0.002)</td>
<td>$0.058$ (0.002)</td>
<td>$0.042$ (0.001)</td>
<td>$0.043$ (0.001)</td>
<td>$0.038$ (0.003)</td>
</tr>
<tr>
<td>Mean ISE(CV)</td>
<td>$-63.196$ (0.963)</td>
<td>$-61.996$ (1.061)</td>
<td>$-61.996$ (1.061)</td>
<td>$-61.996$ (1.061)</td>
<td>$-67.116$ (1.321)</td>
</tr>
<tr>
<td>Mean ISE(KDE)</td>
<td>$-2.659$ (0.073)</td>
<td>$-2.742$ (0.025)</td>
<td>$-2.742$ (0.025)</td>
<td>$-2.742$ (0.025)</td>
<td>$-2.994$ (0.023)</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$-49.267$ (3.001)</td>
<td>$-114.018$ (10.891)</td>
<td>$-114.018$ (10.891)</td>
<td>$-114.018$ (10.891)</td>
<td>$-141.744$ (8.473)</td>
</tr>
<tr>
<td>Mean KL(KDE)</td>
<td>$-1.803$ (0.147)</td>
<td>$-2.038$ (0.149)</td>
<td>$-2.038$ (0.149)</td>
<td>$-2.038$ (0.149)</td>
<td>$-2.640$ (0.125)</td>
</tr>
<tr>
<td>Mean KL(CV)</td>
<td>$-2.013$ (0.121)</td>
<td>$-2.251$ (0.118)</td>
<td>$-2.251$ (0.118)</td>
<td>$-2.251$ (0.118)</td>
<td>$-3.583$ (0.185)</td>
</tr>
</tbody>
</table>

Fig. 4.9 shows the contours plots of three estimated densities. Compared with the two KDE’s, the MDE gives higher peaks and lower valleys, an indication of the deconvolution effect. For $d = 5, 9$ and $13$, the MDE, with $\hat{h} = 0.117$ ($= 0.836$s),
0.186 (= 0.945s) and 0.188 (= 0.964s), had 16, 36 and 71 components. Moreover, all the univariate marginal densities by the MDE for \( d = 5, 9 \) and 13 are shown in Fig. 4.10.

![Figure 4.10: Univariate marginal densities by the MDE for the Wine data.](image)
4.3 Concluding Remarks

In this chapter, we studied both simulated and real-world data set using the non-parametric multivariate mixture-based density estimator proposed in Chapter 3. The new CNMMB algorithm worked quite well for all the problems studied in this chapter. Fig. 4.11(a) shows the rapid monotone increase of the log-likelihood with iteration number $t$, for the bivariate Indian Liver Patient data. The AIC values vs a fine grid of 30 $h$-values for the same data are shown in Fig. 4.11(b).

Figure 4.11: (a) Log-likelihood path for the bivariate Indian Liver Patient data, and (b) the AIC path for various choices of $h$-values.

<table>
<thead>
<tr>
<th>$d$</th>
<th>KDE-PI</th>
<th>KDE-SCV</th>
<th>MDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0:01</td>
<td>0:02</td>
<td>1:31</td>
</tr>
<tr>
<td>4</td>
<td>12:28</td>
<td>13:50</td>
<td>4:46</td>
</tr>
<tr>
<td>6</td>
<td>—</td>
<td>—</td>
<td>9:13</td>
</tr>
<tr>
<td>8</td>
<td>—</td>
<td>—</td>
<td>9:04</td>
</tr>
</tbody>
</table>

Table 4.6: Computation times (mm:ss) for the Indian Liver Patient data. For the MDE, it includes the time needed for all 10 $h$-values defined by the grid evenly spaced from $\sqrt[10]{0.1}s$ to $s$.

All computations were carried out on a computer with a clock-speed of 2.61GHz. Some representative running times are given in Table 4.6, for our MDE method and the two KDE procedures provided in the R package ks, for fitting to the Indian Liver Patient data with 2, 4, 6 and 8 variables, respectively. When $d = 2$, the MDE
procedure was much slower than the two KDE ones. However, when \( d = 4 \), the time needed by a KDE increases dramatically and the MDE requires only about a third of the time needed by a KDE. This relative speedup is more obvious in a higher dimensional space. When \( d = 6 \) or 8, it becomes computationally too costly for the KDE’s to produce solutions in a reasonable time (more than 2 hours). In fact, the two implemented KDE procedures do not allow for \( d > 6 \), presumably because of the formidable computational cost. By contrast, the time needed by the MDE only increases gradually with \( d \).
Chapter 5

Classification Using Semiparametric Mixtures

5.1 Introduction

Classification has a wide range of practical applications. For example, it can be of interest to classify tumors using gene expression data in cancer research (Dudoit et al., 2000), to detect spam emails (Hastie et al., 2009), or to assess the accuracy of the labels for food samples in food authenticity studies (Murphy et al., 2010).

A large number of classification methods have been proposed in the literature and widely used in practice (see, for example, Bishop (2006) and Hastie et al. (2009)). Discriminant analysis may be one of the earliest and most widely known approaches, which includes the linear discriminant analysis (Barnard, 1935; Fisher, 1936), and the quadratic discriminant analysis (see, e.g., Fraley and Raftery (2002) and Hastie et al. (2009)). The $k$-nearest-neighbor method was proposed by Fix and Hodges (1951), which classifies an observation to the class that is the most common among its $k$ nearest observations. Breiman et al. (1984) presented a widely-used classification method, called “classification trees”, where tree-structured models are constructed and each leaf node is designated with a class label. Kernel density estima-
tion can also be employed to estimate each class density, which is known as kernel
discriminant analysis (Scott, 1992). Support vector machines are also popularly
used, which characterize classifiers through decision functions (Vapnik, 2000).

Mixture models also provide satisfactory solutions to classification, which is known
as model-based discriminant analysis (MDA), but most of the relevant work are lim-
ited to finite mixtures. See Chapter 2.6 for some literature review about the MDA.
The MDA enjoys more flexibility and the ability of dealing with complicated bound-
aries between classes compared to the linear and quadratic discriminant analysis.
However, difficulties exist with the use of finite mixtures. The log-likelihood function
may have multiple maxima for a given number of components. For homoscedastic
mixtures, it is almost impossible to guarantee finding the global maximum, espe-
cially in a high dimensional space. For heteroscedastic mixtures, the likelihood
function is unbounded, while the local maximum that is deemed “appropriate” is
not well defined for a finite sample, not to mention how to find it exactly. Fitting
a heteroscedastic mixture is also numerically difficult, because a covariance matrix
can easily become singular.

In this chapter, we propose using semiparametric mixtures for classification. In
particular, a semiparametric mixture distribution is used to provide a density esti-
mate for each class, which will naturally provide the discriminant boundaries. Fit-
ting a semiparametric mixture distribution in the multivariate case has long been
considered extremely difficult, but the method proposed in Chapter 3 has made great
computational progress in the context of density estimation. The empirical studies
in Chapter 4 show that a multivariate mixture-based density estimator performs
significantly better than other nonparametric methods. It is hence worthwhile to
extend this method to classification and investigate the performance of the resulting
classifier. We will call the new classifier “semiparametric-mixture-based-classifier”
and further discuss some issues more related to classification. As in Chapter 3,
we will also only study the situation with continuous variables. Simulation and
real-world data studies show that the new method outperforms other six classifiers that are popularly used in the literature, and remarkably when the sample size is moderate or large.

The remainder of this chapter is organized as follows. In Section 5.2, classification methods based on density estimation are briefly reviewed. Section 5.3 describes the new methodology of applying semiparametric mixtures to classification. Simulation studies using three different models are presented in Section 5.4 and six real-world data sets are analyzed in Section 5.5. Concluding remarks are given in the last section.

5.2 Classification Based on Density Estimation

To classify a new observation is typically based on a probabilistic argument. For example, $k$-nearest-neighbor classifiers and classification trees label the observation according to the majority rule, i.e., the majority class in a region properly chosen by the method to which the observation belongs. Alternatively, one can make use of density estimation and labels the observation using the maximum a posterior (MAP) method, i.e., the class with the highest posterior probability. The posterior probability for an observation $(y, x)$ belonging to class $k$ is given by

$$
\Pr[y = k | x] = \frac{p_k f_k(x)}{\sum_{l=1}^{q} p_l f_l(x)}, \quad (5.1)
$$

where $f_k(x)$ denotes the density function for an observation in class $k$ ($= 1, \ldots, q$) and $p_k$ the proportion of observations of class $k$ in the population. Normality assumptions of these distributions lead to linear or quadratic decision boundaries. However, such boundaries may be overly simple and very much inappropriate in a complicated situation. In such cases, nonparametric density estimators can offer better solutions.
Methods based on finite mixtures, known as model-based methods (McLachlan, 1992; Scott, 1992; Hastie and Tibshirani, 1996; Bensmail and Celeux, 1996; Fraley and Raftery, 2002; Hastie et al., 2009), allow each class density to be a finite mixture of normals, namely

\[ f_k(x) = \sum_{j=1}^{m_k} \pi_{kj} \phi_{kj}(x|\theta_{kj}, \Sigma_{kj}), \]

where \( \pi_{kj} \) is the proportion of the \( j \)th mixture component in the population of class \( k \) and \( \phi_{kj}(x|\theta_{kj}, \Sigma_{kj}) \) the normal density for the \( j \)th component with mean \( \theta_{kj} \) and covariance matrix \( \Sigma_{kj} \). For each class, the maximum likelihood estimate of the mixture is usually computed for each number of mixture components and a model selection method is used to determine a proper number of components. By allowing the component covariance matrices to be identical or vary across components, the mixture is either homoscedastic or heteroscedastic. Model-based classification has been extended by Bensmail and Celeux (1996) by using the eigenvalue decomposition of \( \Sigma_{kj} \):

\[ \Sigma_{kj} = \lambda_{kj} \mathbf{D}_{kj} \mathbf{A}_{kj} \mathbf{D}_{kj}', \]

where \( \lambda_{kj} \) (a scalar) determines the volume of \( \Sigma_{kj} \), \( \mathbf{D}_{kj} \) (with normalized eigenvectors) the orientation, and \( \mathbf{A}_{kj} \) (diagonal with determinant 1) the shape. Different assumptions can be imposed upon the covariance matrix by varying parameters in (5.2).

It is also possible to use kernel density estimates for classification, but it would be increasingly hard as the dimensions increase. In the univariate case, it is only a scalar bandwidth whose value needs to be determined, while in the multivariate case, it is a whole matrix to be estimated.
5.3 Classification

5.3.1 Density estimation for a single class

Instead of finite mixtures, in this thesis we propose using nonparametric and semi-parametric mixtures for classification. The density of a nonparametric normal mixture distribution is given by

\[
f_{\Sigma}(x; G) = \int_{\mathbb{R}^d} \phi(x; \theta, \Sigma) \, dG(\theta),
\]

(5.3)

where \(x, \theta \in \mathbb{R}^d\) and \(G(\theta)\) is the mixing distribution function with a completely unspecified form. Here the covariance matrix \(\Sigma\) is fixed, which plays the role of a bandwidth matrix in kernel density estimation to control the smoothness of the density estimate. We consider the method in Chapter 3, in the context of density estimation, the decomposition \(\Sigma = h^2B\), where \(h = \frac{|\Sigma|^{\frac{1}{2}}}{|B|} \) and \(B\) is symmetric, positive-definite and subject to \(|B| = 1\). This gives a semiparametric mixture formulation as follows,

\[
f_h(x; G, B) = \sum_{j=1}^{m} \pi_j \phi(x; \theta_j, h^2B).
\]

(5.4)

The log-likelihood, given by

\[
l_h(G, B) = \sum_{i=1}^{n} \log \left\{ f_h(x_i; G, B) \right\},
\]

(5.5)

is bounded above by \( -\frac{nd}{2} \log(2\pi h^2) \) for \(h\) fixed. The scalar parameter \(h\) controls only the volume of \(\Sigma\) and its value can be determined via model selection. The hybrid method (CNMMB) proposed in Chapter 3 is used to estimate the density function for a single class by maximizing the log-likelihood function (5.5) with respect for \(G\) and \(B\) for \(h\) fixed.
5.3.2 Volume selection in classification

As pointed out by Friedman (1997), the bandwidth parameter should take a much larger value for kernel discriminant analysis than for kernel density estimation. Following this, we consider using the $\text{AIC}_c$, which is given by

$$\text{AIC}_c(h) = \text{AIC} + \frac{2p(p + 1)}{n - p - 1}$$

(Sugiura, 1978; Hurvich and Tsai, 1989). It penalizes the model complexity more severely than the AIC, by adding an extra correction term, and, as strongly recommended by Burnham and Anderson (2002), is supposed to work better, when the sample size is small or the number of parameters is large, e.g., when $n/p_{\max} < 40$, $p_{\max}$ being the number of parameters for the most complex model among all the candidates. Because of the extra term, the $\text{AIC}_c$ chooses more parsimonious models and hence, for our mixture density estimates, larger $h$-values. We found in our simulation and real-world data studies, the smoother density estimates that resulted form the use of the $\text{AIC}_c$ did in general lower misclassification rates. Because of this general favour of using smoother density estimates, along with the nonparametric and adaptive nature of $G$, it does not seem necessary to consider many potential $h$-values. In our studies, we only used a grid of 5 $h$-values.

5.3.3 Variable selection

In classification, using variable/feature selection to remove redundant variables could lead to parsimonious yet more accurate classifiers (Xiong et al., 2001; Fukumizu et al., 2004; Krishnapuram et al., 2004). This is especially important for high-dimensional datasets which may contain many irrelevant variables. Since our density estimator is based on likelihood, variable selection in principle can also be made via information-theoretic criteria. For example, one may consider using the full log-likelihood of all the classes and proceed with variable selection by backward
elimination, forward selection or stepwise selection, as typically carried out for linear regression problems.

However, such a variable selection procedure via an information-theoretic criterion can be computationally very expensive, especially when there exist a large number of variables. Our new classifier has the strength of more accurate classification but also the weakness of higher computational costs than most others; see the studies below. It is therefore recommended that variables be selected via some other independent procedure. There are, of course, many such procedures available in the literature. In our implementation, we simply use classification trees for finding the most relevant variables and then apply the new classifier in the dimension-reduced variable space. Classifications trees are fast to build but often not very accurate in terms of prediction, largely because of their discrete modelling of class probabilities. Therefore, their strengths and weaknesses complement very much those of our new classifier.

5.3.4 The new classifier

After density estimates are made available for all classes, classification is straightforward based on posterior probabilities. This gives the “semiparametric-mixture-based classifier” (SPMC). We have also considered constructing a SPMC using an identical $h$-value or an identical $B$ across all classes, but the performance generally got worse. This is likely because of the heterogeneity among different classes.

Being a new, strong competitor for solving classification problems, the SPMC produces smooth, curved decision boundaries and can handle well sophisticated situations; see Section 5.4. In comparison, the $k$-nearest-neighbor method tends to produce non-smooth, irregular boundaries. Standard classification trees use linear, axis-parallel boundaries and support vector machines produce smooth but often relatively simple boundaries. The SPMC also has advantages over other density-
based classification methods, owing to the deconvolution nature of a semiparametric mixture and its superior performance in density estimation. A kernel density estimator, being a convolution between the kernel function and the empirical probability mass function, tends to under-estimate around a peak but over-estimate in a valley. Model-based classification methods using finite mixtures have difficulties in fitting mixtures when many components are needed and in finding the global maximum of a likelihood function, which inevitably will reduce their classification accuracy.

5.4 Simulation Studies

5.4.1 Set-up

We carried out simulation and real-world data studies in R (R Core Team, 2013) to investigate the performance of the SPMC and to compare it with six other classification methods: the support vector machines (SVM), the \(k\)-nearest-neighbor (KNN) classifier (for \(k = 10\)), classification trees (RPart), classification based on kernel density estimates (KDE), the homoscedastic (MDA-Ho) and the heteroscedastic mixture method (MDA-He).

For these six other methods, we used their implementations with from the following R packages, respectively: \texttt{e1071} (Meyer et al., 2014), \texttt{class} (Ripley and Venables, 2014), \texttt{rpart} (Therneau et al., 2014), \texttt{ks} (Duong, 2008) and \texttt{mclust} (Fraley et al., 2013). In particular, to build a SVM, two parameters, \(\gamma\) and \(c\), are need to be selected in the \texttt{e1071} and the defaults lead to poor performance. Since there are only two parameters, we considered a grid search to select appropriate values. Exponentially growing sequences of \(\gamma\) and \(c\) are a popular practical method to select the parameters and the grids here are chosen to be \(2^{-5,-4,...,4,5}\) for \(\gamma\) and \(2^{-5,-4,...,4,5}\) for \(c\). Although classification trees are not very good classifier, we still kept it as a competing method since it has strength when it successfully identify the critical
variables as shown in the example of Diabetes in Section 5.5.

For simulation studies reported below, we only considered two-class problems, as the performance of our method should likely carry over to multi-classes problems. Moreover, we will mainly study two-dimensional problems so that their results can be easily visualized and understood; real-world data in various dimensional spaces are to be studied in Section 5.5.

We first consider in Section 5.4.2 using a normal distribution for observations in each class, where each normal distribution has its own mean and covariance matrix and is either skewed or not. In Section 5.4.3, we consider using a complex, nonparametric distribution for each class.

To apply the AIC$_c$ to the SPMC, we considered a grid of 5 potential $h$-values that were evenly distributed between $\sqrt[3]{0.1}s$ and $s$, $s$ being the volume parameter value of the sample covariance matrix for the class. For each class, a density estimate was found by the CNMMB algorithm for each $h$-value, starting with $h = s$, and then AIC$_c$ is used to determine the appropriate the $h$-value and the corresponding mixture density.

### 5.4.2 Classes with normal distributions

In this situation, each of two classes simply has a normal, but possibly skewed, distribution. In particular, the skewed normal distribution used has density of the form (Azzalini and Valle, 1996):

$$f(x; \mu, \Sigma, \alpha) = 2\phi(x; \mu, \Sigma)\Phi(\alpha^\top(x - \mu)),$$

where $\Phi(\cdot)$ the univariate standard normal distribution function. Two dimensionalities, $d = 2$ and 10, were considered. The parameter values in four studied cases are given in Table 5.1.

Each of the seven methods was applied to 100 random repetitions in each case,
Table 5.1: Parameter values

<table>
<thead>
<tr>
<th>Class Distribution (d)</th>
<th>Class 1</th>
<th>Class 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal (2)</td>
<td>( \mu_1 = 0, \Sigma_1 = \begin{pmatrix} 1 &amp; 0.7 \ 0.7 &amp; 1 \end{pmatrix} )</td>
<td>( \mu_2 = (-1.5, -0.5), \Sigma_2 = \begin{pmatrix} 1 &amp; -0.9 \ -0.9 &amp; 1 \end{pmatrix} )</td>
</tr>
<tr>
<td>Skewed normal (2)</td>
<td>Same ( \mu_1, \Sigma_1 ), but ( \alpha = \begin{pmatrix} 0 \ 4 \end{pmatrix} )</td>
<td>Same ( \mu_2, \Sigma_2 ), but ( \alpha = \begin{pmatrix} 0 \ 4 \end{pmatrix} )</td>
</tr>
<tr>
<td>Normal (10)</td>
<td>( \mu_1 = 0, \Sigma_1 = \begin{pmatrix} 1 &amp; 0.2 &amp; 0 &amp; \cdots &amp; 0 \ 0.2 &amp; 1 &amp; 0.2 &amp; \cdots &amp; 0 \ \vdots &amp; \vdots &amp; \vdots &amp; \ddots &amp; \vdots \ 0 &amp; 0 &amp; 0 &amp; \cdots &amp; 1 \end{pmatrix} )</td>
<td>( \mu_2 = -1/2, \Sigma_2 = \begin{pmatrix} 0 &amp; -0.5 &amp; 0 &amp; \cdots &amp; 0 \ -0.5 &amp; 1 &amp; -0.5 &amp; \cdots &amp; 0 \ \vdots &amp; \vdots &amp; \vdots &amp; \ddots &amp; \vdots \ 0 &amp; 0 &amp; 0 &amp; \cdots &amp; 1 \end{pmatrix} )</td>
</tr>
<tr>
<td>Skewed normal (10)</td>
<td>Same ( \mu_1, \Sigma_1 ), but ( \alpha = 4 )</td>
<td>Same ( \Sigma_2 ), but ( \mu_2 = 1/2, \alpha = 4 )</td>
</tr>
</tbody>
</table>

Figure 5.1: Classes with normal distribution, with decision boundaries shown in thick curves.

where each repetition had 350 random observations of class 1 and 150 random observations of class 2. For the 2-dimensional problems, density contour curves, adjusted by numbers of observations, are shown in the upper-left panels of Figures 5.1 and 5.2, along with the true decision boundaries. The other panels show the decision boundaries found by the seven methods from a typical random realization, with observations plotted in “circles” for class 1 and “pluses” for class 2.

As shown in Figures 5.1 and 5.2, all methods can find decision boundaries reasonably similar to the true ones, but they differ in accuracy. Table 5.2 gives their misclassification rates, averaged over the 100 repetitions. The new classifier SPMC has the lowest misclassification rates in the 2-dimensional cases and nearly the lowest
5.4. SIMULATION STUDIES

5.4.3 Classes with nonparametric distributions

To make simulation studies more realistic, we considered generating data from three classification models estimated from a real-world data set, namely the artificial characters data set in the Machine Learning Repository at the University of California at Irvine (Bache and Lichman, 2013). In particular, 200 observations were randomly chosen from the original data set with 100 of letter ‘A’ and 100 of ‘C’. The two predictor variables are the length of a segment and the length of the diagonal of the smallest rectangle. Three different models, labelled Scenarios 1–3, were obtained by KDE, MDA-Ho and SPMC, respectively. As shown in the upper-left panels in Fig-

Figure 5.2: Classes with skewed normal distributions, with decision boundaries shown in thick curves.

misclassification rates in the 10-dimensional cases.

Table 5.2: Misclassification rates (×100%) for classes with normal distributions, with standard errors given in subscripts.

<table>
<thead>
<tr>
<th>Class Distribution (d)</th>
<th>KNN</th>
<th>RPart</th>
<th>SVM</th>
<th>KDE</th>
<th>MDA-Ho</th>
<th>MDA-He</th>
<th>SPMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal (2)</td>
<td>17.3_{0.16}</td>
<td>17.5_{0.36}</td>
<td>18.7_{0.18}</td>
<td>16.2_{0.69}</td>
<td>15.8_{0.15}</td>
<td>15.8_{0.15}</td>
<td>14.3_{0.37}</td>
</tr>
<tr>
<td>Skewed normal (2)</td>
<td>11.3_{0.37}</td>
<td>9.7_{0.44}</td>
<td>8.4_{0.26}</td>
<td>8.5_{0.27}</td>
<td>8.4_{0.16}</td>
<td>8.2_{0.16}</td>
<td>7.6_{0.10}</td>
</tr>
<tr>
<td>Normal (10)</td>
<td>14.8_{0.19}</td>
<td>29.7_{0.62}</td>
<td>10.7_{0.39}</td>
<td>—</td>
<td>5.3_{0.12}</td>
<td>5.6_{0.17}</td>
<td>8.4_{0.22}</td>
</tr>
<tr>
<td>Skewed normal (10)</td>
<td>17.5_{0.28}</td>
<td>31.4_{0.49}</td>
<td>12.5_{0.31}</td>
<td>—</td>
<td>5.4_{0.13}</td>
<td>6.0_{0.26}</td>
<td>5.5_{0.30}</td>
</tr>
</tbody>
</table>
Figure 5.3: For the upper-left, true discriminant boundaries between classes (thick solid curves) and contour lines for the true densities of classes ‘A’ (long-dashed curves) and ‘C’ (thin solid curves) for a data set of size 500, with equal class proportions. For the rest, discriminant boundaries between classes (solid curves) and scatterplot for classes ‘A’ (‘circles’) and ‘C’ (‘pluses’).

Figure 5.4: For the upper-left, true discriminant boundaries between classes (thick solid curves) and contour lines for the true densities of classes ‘A’ (long-dashed curves) and ‘C’ (thin solid curves) for a data set of size 500, with equal class proportions. For the rest, discriminant boundaries between classes (solid curves) and scatterplot for classes ‘A’ (‘circles’) and ‘C’ (‘pluses’).
Figure 5.5: For the upper-left, true discriminant boundaries between classes (thick solid curves) and contour lines for the true densities of classes ‘A’ (long-dashed curves) and ‘C’ (thin solid curves) for a data set of size 500, with equal class proportions. For the rest, discriminant boundaries between classes (solid curves) and scatterplot for classes ‘A’ (“circles”) and ‘C’ (“pluses”).

In Figures 5.3–5.5, the true discriminant boundaries have an increasing level of complexity in the three scenarios. Two sample sizes, \( n = 100 \) and 500, with two proportions, 50\% and 40\%, for class ‘A’, were considered and 100 data sets were generated in each case. Figures 5.3–5.5 also show the discriminant boundaries found by the seven classifiers for a generic random sample of size 500 with equal class proportions. It can be seen that the density-based methods (KDE, MDAs and SPMC) handles well both simple and complicated discriminant boundaries. The RPart and SVM approximate simple boundaries well in Scenario 1, but they have increasing difficulties in finding accurate boundaries as the true boundaries become more complicated. The boundaries found by the KNN are fairly poor in all three scenarios.

Table 5.3 provides a summary of the simulation study results. It gives the averaged misclassification rates, along with standard errors, of all investigated classification methods in each case. It is clear that the density-based methods generally performed better than the other methods. The SPMC outperformed, often significantly, the
Table 5.3: Misclassification rates (×100%) for 3 scenarios with different sample sizes or class proportions, with standard errors given in subscripts.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>n = 100 (= 50 + 50)</th>
<th>n = 100 (= 40 + 60)</th>
<th>n = 500 (= 250 + 250)</th>
<th>n = 500 (= 200 + 300)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36.00.79 35.40.64 34.80.39 33.30.77 36.20.66 33.20.81 34.10.82</td>
<td>36.60.74 38.51.36 38.30.68 35.50.77 38.80.66 33.10.70 33.70.42</td>
<td>29.60.24 31.40.59 28.40.69 27.20.19 33.60.24 27.60.30 26.30.14</td>
<td>31.20.19 31.50.35 30.90.39 29.80.26 38.50.60 28.20.35 27.30.11</td>
</tr>
<tr>
<td>2</td>
<td>39.20.54 39.40.50 39.12.34 37.50.54 36.90.33 38.80.80 37.80.70</td>
<td>41.10.89 44.01.41 41.81.43 40.00.89 39.70.86 39.70.77 39.40.79</td>
<td>37.70.38 38.30.53 35.60.11 35.50.16 36.40.11 36.20.19 35.50.15</td>
<td>38.50.36 38.60.60 37.20.27 37.20.31 37.50.30 37.30.36 36.70.18</td>
</tr>
<tr>
<td>3</td>
<td>36.90.79 36.10.99 34.10.41 33.50.60 33.60.35 34.10.78 33.40.83</td>
<td>37.90.80 37.61.44 38.70.87 36.31.13 36.80.83 36.21.22 34.00.71</td>
<td>31.80.26 30.50.17 30.00.33 29.60.26 33.20.11 27.80.28 27.30.17</td>
<td>32.20.31 31.60.53 32.60.27 31.40.19 35.60.46 28.20.21 27.60.12</td>
</tr>
</tbody>
</table>

other six methods and was always superior when the sample size is large. It can also be seen that its outperformance is more obvious when classes have unequal proportions.

5.5 Real-world Data Sets

We have also studied 9 real-world date sets. In particular, the artificial characters, the Pima Indians diabetes, and the breast cancer Wisconsin diagnostic data were obtained from the Machine Learning Repository at the University of California at Irvine (Bache and Lichman, 2013), the banana and the phoneme data from the Knowledge Extraction based on Evolutionary Learning data (Alcalá-Fdez et al., 2011), the dystrophy and the GlaucomaMVF data from the R package ipred, and the diabetes and the banknote data from the R package mclust.

To help reduce the dimensionality of a classification problem when there exist many variables so that the SPMC can be used more effectively (Section 5.3.3), we used RPart to find the most relevant variables, i.e., the variables in the pruned tree that is determined by cross-validation. The resulting dimension-reduced version of
SPMC is denoted by SPMCD.

These 9 data sets are briefly described below, and the pairwise scatter plots for all the data sets are also provided where the observations of the majority class are plotted in red and those of the minority class in blue.

Chars

The original characters data was artificially generated by using a first-order Horn clauses theory, which describes the structure of ten capital letters of the English alphabet, A, C, D, E, F, G, H, L, P and R. Each letter is represented as a set of segments that are described by two variables, the size and the diag. In details, the size is the length of a segment computed by using the geometric distance between two points and the diag is the length of the diagonal of the smallest rectangle which includes the picture of the character.

Some of these capital letters are shown in Figures 5.6.

Figure 5.6: Some letters corresponding to some instances from Chars.
We consider Chars data set with different number of classes here. Firstly, we consider two classes of letters A and C and 1431 observations here, where letters A contains 732 observations and letters C contains 699 observations. Scatterplot of the Chars with two classes is shown in the upper-left in Figures 5.7. Figures 5.7 also shows the discriminant boundaries found by the seven classifiers. It can be seen that the two classes in this dataset are seriously overlapped, where the KNN and the RPart are not able to find the boundaries accurately compared with the other five competitors.

Chars with 5 classes contains letters A, C, D, E and F, where class with letters D has 600 observations, class with letters E 600 observations and class with letters F has 600 observations. Moreover, Chars with 10 classes contains all the letters, where class with letters G, H, L, P and R has 600 observations, respectively.

![Figure 5.7: Scatterplot for classes ‘A’ (“circles”) and ‘C’ (“pluses”) and discriminant boundaries between classes (solid curves).](image)

**Banana**

It is an artificial data set which contains two classes and 5300 observations. Observations of the Banana belongs to several clusters with a banana shape, where class
1 contains 2376 observations and class 2 contains 2924 observations. Two variables, At1 and At2, corresponds to the $x$- and $y$-axis, respectively.

Figures 5.8 shows a scatterplot of Banana, along with the discriminant boundaries found by seven classifiers. We can see that the discriminant boundaries between the two classes are reasonably clear and almost all the classifiers are able to discriminate them.

![Scatterplot of Banana and discriminant boundaries](image)

Figure 5.8: Scatterplot of Banana for class1 ("circles") and 2 ("pluses") and discriminant boundaries between classes (solid curves).

**Diabetes**

The Diabetes contains 145 observations and 3 variables. The 3 variables represents three measurements from clinical diagnosis for diabetes study which are Glucose, Insulin and Sspg. Two classes are included, normal and diabetic, where the normal group contains 76 observations and the diabetic group contains 69 observations. The aim of this data set is to distinguish the diabetic from a normal case.

Figures 5.9 shows the pairwise scatter plots for this data set. The normal are drawn in red and "circles" and the diabetic are drawn in blue and "pluses". It can been see that the normal group gather together and the diabetic spread out.
Moreover, the boundaries between the two classes are clear by Insulin.

![Pairwise scatter plots for Diabetes.](image)

**Figure 5.9: Pairwise scatter plots for Diabetes.**

**Phoneme**

The Phoneme contains 5404 observations and 5 variables. Two classes are included, the nasal and the oral sounds, where the nasal contains 3818 observations and the oral contains 1586 observations. The aim of this dataset is to distinguish between nasal and oral sounds. Each observation in the Phoneme is described by five variables, the Sh as in she, the Dcl as in dark, the Ly as the vowel in she, the Aa as
the vowel in dark, and the Ao as the first vowel in water. Figures 5.10 shows the pairwise scatter plots of the Phoneme. In Figures 5.10, observations of the nasal class are shown in red and “circles” and those of the oral class are shown in blue and “pluses”.

**Figure 5.10: Pairwise scatter plots for Phoneme.**

**Banknote**

The Banknote contains 200 bank notes with two classes and six measurements. In details, there are 100 genuine and 100 counterfeit old-Swiss 1000-franc bank notes.
The aim of this data set is to identify the fraudulent. Each bank note is measured by six variables which are listed as follows.

**Length** Length of a bill.

**Left** Width of the left edge of a bill.

**Right** Width of the right edge of a bill.

**Bottom** Bottom margin width of a bill.

**Top** Top margin width of a bill.

**Diagonal** Length of the diagonal of a bill.

Figures 5.11 shows the pairwise scatter plots of the Banknote. The fraudulent are shown in red and “circles” and the genuine are shown in blue and “pluses”. It can be seen that the two groups can be well separated by some variables.

**Dystrophy**

Duchenne muscular dystrophy (DMD) is a genetically transmitted disease, passed from a mother to her children. The aim of this data set is to detect the muscular dystrophy carriers. The Dystrophy contains 209 observations with two classes, the female DMD non-carriers and the female DMD carriers, where the non-carrier group contains 134 observations and the carrier group contains 75 observations. Six variables are included, and listed as follows.

**Age** Age in years.

**M** Month of examination.

**CK** Serum marker creatine kinase.

**H** Serum marker hemopexin.
5.5. REAL-WORLD DATA SETS

Figure 5.11: Pairwise scatter plots for Banknote.

**PK** Serum marker pyruvate kinase.

**LD** Serum marker lactate dehydrogenase.

Figures 5.12 shows the pairwise scatter plots of the Dystrophy. The female DMD non-carriers are shown in red and “circles” and the female DMD carriers are shown in blue and “pluses”. The aim of this data set is to identify the female DMD carriers. It can be seen that the DMD carriers have a larger spread than the DMD non-carriers.
Pima

The Pima Indian diabetes data set contains 768 observations with 6 variables. Two classes are included, those normal and those tested positive for diabetes, based on World Health Organization criteria. The normal group contains 500 observations and the group of diabetes contains 268 observations. The aim of this data set is to classify females at least 21 years old of Pima Indian heritage as diabetic or not. Six variables are listed as follows.

Plas Plasma glucose concentration of an oral glucose tolerance test.
5.5. REAL-WORLD DATA SETS

**Pres** Diastolic blood pressure.

**Skin** Triceps skin fold thickness.

**Insu** 2-Hour serum insulin.

**Mass** Body mass index.

**Pedi** Diabetes pedigree function.

**Age** Age in years.

Figures 5.13 shows the pairwise scatter plots of the Pima. The normal are shown in red and “circles” and the diabetic in blue and “pluses”.

**Cancer**

The Breast cancer Wisconsin diagnostic data set contains 569 observations with 10 variable, and two classes, the benign and the malignant. The benign contains 357 observations and the malignant contains 212 observations. This data set describes characteristics of the cell nuclei present in the image which are computed from a digitized image of a fine needle aspirate of a breast mass. 10 variables are described as follows.

**radius** The distances from center to points on the perimeter.

**Texture** The standard deviation of gray-scale values.

**Perimeter**

**Area**

**Smoothness** The local variation in radius lengths.

**Compactness** $\text{Perimeter}^2/\text{area} - 1.0.$
Figure 5.13: Pairwise scatter plots for Pima.

**Concavity** The severity of concave portions of the contour.

**Concave points** The number of concave portions of the contour.

**Symmetry**

**Fractal dimension** “coastline approximation” - 1.

Figures 5.14 shows the pairwise scatter plots of the Cancer. The benign are shown in red and “circles” and the malignant are shown in blue and “pluses”. The two classes are well separated by many variables, as shown in the plots.
5.5. REAL-WORLD DATA SETS

GlaucomaMVF

The GlaucomaMVF data set contains 170 observations with 66 variables and two classes, the glaucoma and the normal. The glaucoma contains 85 observations and the normal also contains 85 observations. 66 variables are derived from a confocal laser scanning image of the optic nerve head, and most of them describe either the area or volume in certain parts of the papilla. Since the number of variables is very big, we do not give their details here.

We summarized the classification results of above data sets in Table 5.4, where
the estimated misclassification rates of various methods, along with their standard errors, based on 100 repetitions of 10-fold cross-validation are given. Some information is also given for each data set, including the sample size \( n \), the dimensionality \( d \) and the number of classes \( k \). Some entries are shown by “—”, either because they are computationally too costly, or because, in the case of the SPMCD, they are the same as those given by the SPMC.

From the pairwise scatter plots shown in Figures 5.7–5.14, we can see that Chars, Phoneme, and Pima all have two seriously overlapping classes and this can be seen in every pairwise plot. While for Banana, Diabetes, Dystrophy, and Cancer, the two classes can be well separated by some variables. For Banknote, the two classes can even be separated by a single variable, namely Diagonal. Comparing the classification results in Table 5.4, it can be seen that the seven classifiers all perform well for the well separated data set, Banknote. Their performance is relatively bad for Chars, Phoneme and Pima, and the MDE has the lowest misclassification rate for Chars and Pima while the SVM has the lowest misclassification rate for Phoneme. For the data sets with some variables where the two classes are well separated, the MDE always has the lowest misclassification rate. One interesting result is for Diabetes, RPart performs quite well because the variable of Insulin separates the two classes very well.

Overall, the results show that the SPMC had very good performance when variable selection is not an issue. If dimensionality reduction is necessary, then the SPMCD outperformed the other methods in almost all cases. It is worth pointing out that in general, density-based methods (KDE, MDAs and SPMC) performed better than the others. We note that the KDE is computationally very demanding in a high-dimensional space and we were unable to obtain results for \( d \geq 5 \).
5.6. CONCLUDING REMARKS

Table 5.4: Misclassification rates (×100%) for real data sets, with standard errors given in subscripts.

<table>
<thead>
<tr>
<th>Data set (n, d, k)</th>
<th>KNN</th>
<th>RPart</th>
<th>SVM</th>
<th>KDE</th>
<th>MDA-Ho</th>
<th>MDA-He</th>
<th>SPMC</th>
<th>SPMCD (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chars (1431, 2, 2)</td>
<td>51.90_0.22</td>
<td>53.90_0.16</td>
<td>37.90_0.06</td>
<td>35.80_0.07</td>
<td>40.90_0.08</td>
<td>35.60_0.22</td>
<td>34.90_0.14</td>
<td>—</td>
</tr>
<tr>
<td>Chars (3231, 2, 5)</td>
<td>73.00_0.11</td>
<td>70.80_0.06</td>
<td>59.50_0.05</td>
<td>58.40_0.05</td>
<td>63.00_0.09</td>
<td>59.90_0.04</td>
<td>58.00_0.04</td>
<td>—</td>
</tr>
<tr>
<td>Chars (6231, 2, 10)</td>
<td>86.70_0.08</td>
<td>86.00_0.03</td>
<td>76.90_0.04</td>
<td>75.90_0.06</td>
<td>78.80_0.07</td>
<td>76.90_0.09</td>
<td>75.30_0.04</td>
<td>—</td>
</tr>
<tr>
<td>Banana (5300, 2, 2)</td>
<td>11.40_0.00</td>
<td>15.70_0.06</td>
<td>9.50_0.05</td>
<td>9.30_0.10</td>
<td>9.60_0.05</td>
<td>9.30_0.01</td>
<td>8.90_0.12</td>
<td>—</td>
</tr>
<tr>
<td>Diabetes (145, 3, 2)</td>
<td>6.50_0.00</td>
<td>2.80_0.00</td>
<td>5.30_0.00</td>
<td>6.20_0.00</td>
<td>9.80_0.00</td>
<td>4.70_0.00</td>
<td>4.60_0.00</td>
<td>2.50_0.00 (2)</td>
</tr>
<tr>
<td>Phoneme (5404, 5, 2)</td>
<td>10.60_0.03</td>
<td>18.20_0.16</td>
<td>10.30_0.04</td>
<td>—</td>
<td>21.30_0.01</td>
<td>18.90_0.16</td>
<td>16.10_0.09</td>
<td>—</td>
</tr>
<tr>
<td>Banknote (200, 6, 2)</td>
<td>0.50_0.00</td>
<td>1.50_0.00</td>
<td>1.40_0.10</td>
<td>—</td>
<td>0.50_0.00</td>
<td>0.50_0.00</td>
<td>0.10_0.00</td>
<td>0.50_0.00 (2)</td>
</tr>
<tr>
<td>Dystrophy (209, 6, 2)</td>
<td>11.60_0.13</td>
<td>14.40_0.13</td>
<td>9.90_0.19</td>
<td>—</td>
<td>9.90_0.15</td>
<td>9.80_0.28</td>
<td>9.90_0.07</td>
<td>9.70_0.09 (4)</td>
</tr>
<tr>
<td>Pima (768, 7, 2)</td>
<td>30.20_0.09</td>
<td>24.70_0.29</td>
<td>23.20_0.17</td>
<td>—</td>
<td>25.40_0.14</td>
<td>29.70_0.24</td>
<td>29.30_0.16</td>
<td>22.70_0.09 (4)</td>
</tr>
<tr>
<td>Cancer (569, 10, 2)</td>
<td>8.50_0.09</td>
<td>9.20_0.15</td>
<td>5.10_0.04</td>
<td>—</td>
<td>4.10_0.08</td>
<td>4.90_0.10</td>
<td>3.70_0.10</td>
<td>4.90_0.07 (4)</td>
</tr>
<tr>
<td>GlaucomaMVF (153, 66, 2)</td>
<td>7.40_0.15</td>
<td>11.10_0.37</td>
<td>13.50_0.67</td>
<td>—</td>
<td>28.50_0.42</td>
<td>32.70_0.59</td>
<td>—</td>
<td>7.20_0.16 (3)</td>
</tr>
</tbody>
</table>

5.6 Concluding Remarks

As presented above, a new classification method is proposed, which extends the nonparametric density estimation method based on semiparametric mixtures that was studied in Chapters 3 and 4; see also Wang and Wang (2015). In particular, the density for the observations in each class is estimated using a homoscedastic mixture yet with an unrestricted number of components, and new observations are classified according to the MAP principle. The method is best applied in a low- or moderate-dimensional space. This means that when there exist a large number of predictor variables, one should better resort to a variable/feature selection method to find relevant ones first. We recommend classification trees for a pre-selection of variables, since it complements well the strengths and weaknesses of the new classifier. Simulation and real-world data studies show that the proposed method generally performs better than six other commonly-used ones. While all classifiers based on density estimation performed favorably, the new classifier gave the lowest, or nearly the lowest, misclassification rates in all cases studied.

As pointed out earlier, a weakness of the new classifier SPMC is its computation cost. Table 5.5 gives the computation times for building a classifier from each full data set. Since the SPMC is implemented in R (R Core Team, 2013), we anticipate a substantial reduction in training time if it is coded in compilable languages such as C or FORTRAN.
Table 5.5: Computation times (in seconds) for building a single classifier from each data set.

<table>
<thead>
<tr>
<th>Data set (n, d, k)</th>
<th>SPMC</th>
<th>Data set (n, d, k)</th>
<th>SPMC</th>
<th>SPMCD</th>
<th>Data set (n, d, k)</th>
<th>SPMC</th>
<th>SPMCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chars (2031, 2, 3)</td>
<td>230</td>
<td>Diabetes (145, 3, 2)</td>
<td>7</td>
<td>3</td>
<td>Dystrophy (209, 7, 2)</td>
<td>89</td>
<td>19</td>
</tr>
<tr>
<td>Chars (5231, 2, 5)</td>
<td>370</td>
<td>Kyphosis (81, 3, 2)</td>
<td>6</td>
<td>2</td>
<td>Pima (768, 8, 2)</td>
<td>335</td>
<td>60</td>
</tr>
<tr>
<td>Chars (6231, 2, 10)</td>
<td>720</td>
<td>Phoneme (5404, 5, 2)</td>
<td>1218</td>
<td>—</td>
<td>Cancer (569, 10, 2)</td>
<td>289</td>
<td>99</td>
</tr>
<tr>
<td>Banana (5300, 2, 2)</td>
<td>386</td>
<td>Banknote (200, 6, 2)</td>
<td>75</td>
<td>16</td>
<td>GlaucomaMVF (153, 66, 2)</td>
<td>—</td>
<td>11</td>
</tr>
</tbody>
</table>

With some evidence provided in our studies, building classifiers based on density estimation can be advantageous. Our results show that classification depends, to a significant extent, on accurate density estimation, which then naturally provides smooth decision boundaries. With density estimates available, it is fairly straightforward to incorporate misclassification costs, if they are available, and is thus able to find cost-dependent decision boundaries with virtually no extra work. Since discrete and categorical variables are common in real-world applications, it is important that the new classifier be extended to deal with them. We shall look into these problems in future research.
6.1 Introduction

Generalized linear mixed models (GLMM’s) extend generalized linear models by involving random effects to the linear predictor and using them to account for within-subject correlation. These models have been widely used for repeated measurements and clustered data, and their applications range from animal breeding programme (Muir, 2005), to patient-specific rates of disease progression (Zhang et al., 2008), and to complex survey data analysis in epidemiology (Brumback et al., 2013).

Fitting GLMM’s typically assumes that the random effects have a normal distribution. The covariance matrix (or variance in the univariate case) of the normal distribution can be estimated from the data and several estimation methods have been developed for it (Laird and Ware, 1982; Breslow and Clayton, 1993; Hedeker and Gibbons, 1994). The normality assumption brings about some computational convenience, especially for linear mixed models, and seemingly also provides a certain level of robust estimation of the fixed effects (Neuhaus et al., 1992, 1994; Butler and Louis, 1997). However, the normality assumption may fail to hold true in real-
ity and if so, the misspecification of the random effects distribution can compromise the estimation efficiency to a significant extent in certain situations (Tao et al., 1999). Neuhaus et al. (1992) showed that the misspecification of the random intercept distribution can lead to biased intercept estimators and Neuhaus et al. (2013) demonstrated that the misspecification of the distribution of random slopes can also result in biased slope estimators.

Owing to their flexibility nature, mixture models have also been used in GLMM’s to protect against misspecification of random effects distribution. Stiratelli et al. (1984) proposed to use a finite mixture of normal distributions to approximate the random effects distribution in longitudinal studies. Verbecke and Lesaffre (1996) demonstrated that random effects could be badly estimated under the normality assumption if random effects are generated from a finite mixture of normal distributions. However, these studies focus on using finite mixtures of normal distributions, which essentially are still parametric models. One can leave the distribution of random effects completely unspecified and let the maximum likelihood determine its estimate, along with the estimates of fixed-effects coefficients, which thus results in a semiparametric mixture formulation. Owing to computational difficulties, most previous studies focus on the univariate case (Aitkin, 1999; Wang, 2010), in particular, for a random intercept. Recently, Lesperance et al. (2014) extend the algorithms proposed by Wang (2007, 2010) to deal with multiple random effects, with numerical studies focusing on the bivariate case, i.e., a random intercept and a random slope. However, the methods of Lesperance et al. (2014) have some problems, which will be explained in Section 6.6.

In this chapter, we study using semiparametric multivariate mixtures for fitting GLMM’s with multiple random effects. The goal is to find the nonparametric MLE for the joint distribution of multiple random effects and the MLE for fixed effects. It has long been considered computationally difficult to find a multivariate nonparametric MLE, but the method CNMMB proposed in Chapter 3 would be instrumental
6.2 GENERALIZED LINEAR MIXED MODELS

for fitting the resulting semiparametric mixture, if it can be modified properly. In the study reported below, we will take logistic regression models as an example and present a new method to computing the corresponding semiparametric MLE. Since Chapter 3 only deals with mixtures of multivariate normal distributions, modifications for the logistic regression model are necessary. Simulated studies show that this semiparametric formulation outperforms significantly the parametric method that is under the normality assumption, in terms of both estimating the random and fixed effects, especially in situations when covariates are highly correlated and random effects follow a non-normal distribution. Real-world data demonstrates that without using normality assumption the new method gives similar estimates of the fixed effects to the traditional method, yet with greater likelihood values.

The remainder of the chapter is organized as follows. In Section 6.2, generalized linear mixed models, especially logistic regression mixed models, are briefly introduced. Section 6.3 describes how to apply semiparametric multivariate mixtures to fit GLMM’s and Section 6.4 an algorithm for finding the nonparametric MLE of the joint distribution of random effects, as well as the MLE of fixed effects. Section 6.5 shows demonstrations of finding an NPMLE and a joint MLE using the proposed algorithm, respectively. The direct search directional derivative method for estimating GLMM’s is described in Section 6.6. Section 6.7 investigates the difference between the proposed method and the direct search directional derivative method for fitting semiparametric mixtures. Concluding remarks are given in the last section.

6.2 Generalized Linear Mixed Models

Generalized linear mixed models are a natural extension of generalized linear models to deal with clustered data by using random coefficients. For clustered data, let us assume that there are $n_i$ observations from the $i$th cluster, $i = 1, \ldots, n$, and that the $j$th observation of cluster $i$ has random-effects covariates $x_{ij}$ and fixed-effects
covariates \( z_{ij} \). Let the conditional joint density of \( y_i = (y_{i1}, \ldots, y_{in_i})^\top \) be given by

\[
f(y_i | X_i, Z_i, \theta_i, \beta) = \prod_{j=1}^{n_i} f(y_{ij} | x_{ij}, z_{ij}, \theta_i, \beta), \tag{6.1}
\]

where \( X_i = (x_{i1}, \ldots, x_{in_i})^\top \), \( Z_i = (z_{i1}, \ldots, z_{in_i})^\top \), and \( \theta_i \) and \( \beta \) are the random and fixed effects, respectively. Let

\[
\mu_{ij} = E[y_{ij} | x_{ij}, z_{ij}, \theta_i, \beta],
\]

and for a linear predictor, the covariate vectors are connected to \( \mu_{ij} \) through a link function

\[
g(\mu_{ij}) = x_{ij}^\top \theta_i + z_{ij}^\top \beta. \tag{6.2}
\]

The study in this chapter focuses on using the binomial distribution for \( f \) on the right-hand side of (6.1) and the logistic link

\[
\logit(\mu_{ij}) = \log\left(\frac{\mu_{ij}}{1 - \mu_{ij}}\right). \tag{6.3}
\]

Therefore the \( j \)th response in the \( i \)th cluster, \( y_{ij} \), has a Binomial \((n_{ij}, p_{ij})\) distribution and link function (6.2) becomes

\[
\logit(p_{ij}) = a_i + \sum_u x_{iju}b_{iu} + \sum_q z_{ijq}\beta_q, \tag{6.4}
\]

where \( a_i \) and \( b_{iu} \) are, respectively, the random intercept and slopes for the \( i \)th cluster, and \( \mu_{ij} = p_{ij} \). Therefore, \( x_{ij} = (1, x_{ij1}, \ldots, x_{iju})^\top \) and \( z_{ij} = (z_{ij1}, \ldots, z_{ijq})^\top \). The conditional density of \( y_i \) thus becomes

\[
f(y_i | X_i, Z_i, \theta_i, \beta) = \prod_{j=1}^{n_i} p_{ij}^{y_{ij}}(1 - p_{ij})^{n_{ij} - y_{ij}}, \tag{6.5}
\]
where \( p_{ij} = \logit^{-1}(x_{ij}^\top \theta_i + z_{ij}^\top \beta) \). If \( n_{ij} = 1 \), as in many practical cases, it reduces to the Bernoulli distribution.

Let the random effects \( \mathbf{\theta}_i = (a_i, \mathbf{b}_i^\top)^\top \) have a joint distribution \( G \). The unconditional density of \( y_i \) is given by

\[
f(y_i|X_i, Z_i, G, \beta) = \int_\Omega f(y_i|X_i, Z_i, \mathbf{\theta}, \beta) \, dG(\mathbf{\theta}),
\]

and the log-likelihood function is

\[
l(G, \beta) = \sum_{i=1}^n \log \left\{ \int_\Omega f(y_i|X_i, Z_i, \mathbf{\theta}, \beta) \, dG(\mathbf{\theta}) \right\}.
\]

This simply means that a GLMM is just a mixture model. The likelihood approach hence gives the maximum likelihood estimate \( (\hat{G}, \hat{\beta}) \) of \( (G, \beta) \). Traditionally, \( G \) is assumed to be a normal distribution. When \( G \) is completely unspecified, the GLMM becomes a semiparametric mixture and we need to find the nonparametric MLE for the infinite-dimensional parameter \( G \) and the MLE for the finite-dimensional parameter \( \beta \).

### 6.3 A Semiparametric Mixture

When \( G \) is completely unspecified, the density function given in (6.6) is in fact the density of a semiparametric mixture model. For every fixed \( \beta \), there always exists a discrete NPMLE \( \hat{G} \) of \( G \), with no more support points than the number of distinct values in the sample. Let a discrete \( G \) be given by

\[
G(\mathbf{\theta}) = \sum_{j=1}^m \pi_j \delta_{\mathbf{\theta}_j},
\]
where \( \pi_j > 0 \), \( \sum_{j=1}^{m} \pi_j = 1 \) and \( \delta_{\theta_j} \) denotes the degenerate distribution function at \( \theta_j \). Density (6.6) thus becomes

\[
f(y_i|X_i, Z_i, \pi, \Theta, \beta) = \sum_{j=1}^{m} \pi_j f(y_i|X_i, Z_i, \theta_j, \beta),
\]

(6.8)

where \( \Theta = (\theta_1, \ldots, \theta_m)^\top \), and log-likelihood function (6.7) becomes

\[
l(\pi, \Theta, \beta) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{m} \pi_j f(y_i|X_i, Z_i, \theta_j, \beta) \right\}.
\]

A special directional derivative of the log-likelihood function is very useful for characterizing \( \hat{G} \), for \( \beta \) fixed. Also known as the gradient function, it is given by

\[
d(\theta; G, \beta) = \lim_{\varepsilon \to 0} \frac{l((1-\varepsilon)G + \varepsilon \delta_{\theta}, \beta) - l(G, \beta)}{\varepsilon}
= \sum_{i=1}^{n} \frac{f(y_i|X_i, Z_i, \theta, \beta)}{f(y_i|X_i, Z_i, G, \beta)} - n.
\]

(6.9)

For any given \( \beta \), it characterizes \( \hat{G} \), in the sense that the following three statements are equivalent:

1. \( \hat{G} \) maximizes \( l(G, \beta) \);
2. \( \hat{G} \) minimizes \( \sup_{\theta} \{d(\theta; G, \beta)\} \);
3. \( \sup_{\theta} \{d(\theta; \hat{G}, \beta)\} = 0 \).

For a semiparametric MLE \( (\hat{G}, \hat{\beta}) \), necessarily it holds that

\[
\left( \frac{\partial l}{\partial \beta} \right)(\hat{G}, \hat{\beta}) = 0.
\]

### 6.4 Computation

Wang (2010) proposed three general methods to computing the maximum likelihood
estimates for fitting a semiparametric mixture model that has a univariate $G$. In this chapter, we will extend one of these methods, the CNM-MS, to the multivariate case. In each iteration, the CNM-MS uses one iteration of the constrained Newton method (CNM) (Wang, 2007) to update $G$ nonparametrically, and then the quasi-Newton BFGS method (see, e.g., (Fletcher, 1987)) to update $(G, \beta)$ where $G$ has a finite support. However, computing a multivariate nonparametric MLE is much more challenging than in the univariate case. The main difficulty with applying the CNM lies in that it is computationally very expensive to find all the local maxima of the gradient function in the multivariate case. Chapter 3 presented a computationally efficient algorithm (CNMM) to find a multivariate MLE. In each iteration, it modifies the CNM by using a random grid to help roughly locate the local maxima of the gradient function, and uses the expectation-maximisation (EM) algorithm to fit a finite normal mixture under the restrictions that $|\mathcal{B}| = 1$. Here, since there is no need to update the covariance matrix and the EM algorithm converges more slowly than the BFGS method, we switch it back to the BFGS for fitting a finite mixture. Therefore, the basic structure of the new algorithm follows more closely that of the CNM-MS of Wang (2010) as reviewed in Section 2.7, while locating roughly the local maxima of the gradient function will resort to the use of a random grid, as done in the CNMM. We name this new algorithm as the CNMM-MS, where the CNMM-MS stands for the multivariate CNM method for fitting semiparametric mixtures.

### 6.4.1 Updating $G$

For any given $\beta$, the gradient function is very helpful to locate new support points since it specifies the deviation from an iterate to $\hat{G}$. To estimate $\hat{G}$, in the univariate case, the CNM adds all the local maxima of the gradient function to the support point set iteratively, but finding all the local maxima becomes computationally expensive in the multivariate case. Here we consider finding the rough locations of
support points by using a grid for gradient evaluation, and let the exact locations be determined by an unconstrained optimization method, here the BFGS.

To be more effective in a multivariate space, we again consider using a random grid instead of a deterministic grid. If the method described in Chapter 3 is directly adopted, such a random grid would be generated from the distribution with density

\[ d^*(\theta; G, \beta) \propto \sum_{i=1}^{n} w_i f(y_i | X_i, Z_i, \theta, \beta), \tag{6.10} \]

where \( f(y_i | X_i, Z_i, \theta, \beta) \) is a mixture component density and \( w_i = f(y_i | X_i, Z_i, G, \beta)^{-1} \).

Note that it is in fact the gradient function (6.9) with the constant term ignored. Sampling based on function (6.10) will give us a random sample with higher density in the areas with larger gradient values. In Chapter 3, such a direct sampling for \( \theta \) from (6.10) is fine, since the component is a normal distribution and hence function (6.10) turns out to an \( n \)-component normal mixture density of \( \theta \). However, for GLMM’s, function (6.10) can not be turned into a proper density function by standardization since \( f(y_i | X_i, Z_i, \theta, \beta) \) may not integrate to a finite value with respect to \( \theta \), as is true for the binomial distribution given in (6.5). We need to truncate the domain of function (6.10) by considering only the \( \theta \)-values such that all \( f(y_i | X_i, Z_i, \theta, \beta) \) are within a certain range. In particular, we choose this range to be between 0.99 and 0.01, since the \( \theta \)-values out of this range contribute too less to the gradient function which could be ignored. Function (6.10) is hence set to 0 for \( \theta \)-values outside the range. There are two good reasons for doing so. One is that the final support points are almost always in this range. The other is that any support point outside the range can be replaced by a point on the boundary of the range with almost a negligible effect on the likelihood value.

Furthermore, we consider obtaining a random grid from two random samples. The first sample is generated from distribution (6.10) that is truncated as described above, by using a Markov chain Monte Carlo (MCMC) method. This random sample
provides a wide coverage of the interesting area, but sometimes may be too sparse to be effective. To provide a better coverage of the areas with higher gradient values, we also include a random sample which is generated compactly around each current support point.

The first random sample is generated by Markov chain Monte Carlo (MCMC) methods based on a truncated distribution (6.10). The MCMC methods are a class of Monte Carlo methods that are quite useful for sampling from probability distributions for which direct sampling is difficult. A random sample corresponds to a Markov chain that is constructed to have the desired distribution as its stationary distribution. We use the widely-used Metropolis-Hastings algorithm, which was first proposed by Metropolis et al. (1953) and later generalized by Hastings (1970). Generally, the Metropolis-Hastings algorithm consists of two steps, with a starting point $x_0$:

1. propose $z_n$ with density $q(z_n|x_n)$,

2. accept $z_n$ with probability $\min\left\{1, \frac{f(z_n)q(x_n|z_n)}{f(x_n)q(z_n|x_n)} \right\}$; otherwise, set $x_{n+1}$ to $x_n$,

where $q(z|x)$ is the proposal density function. In our case, function (6.10) with truncation is the stationary distribution, and the mean of the current support points is chosen as the starting point.

Normally, the proposal distribution $q(z|x)$ should have enough dispersion to lead to an exploration of the entire support of the stationary distribution. Here we use a multivariate normal distribution as the proposal distribution $q$, with mean $\mathbf{0}$ and a covariance matrix which is obtained from the current support points. When there is only one support point, this covariance matrix obtained from the random grid in the previous iteration is used again. The determinant value of this covariance matrix is called step size. For a given sample size, a big step size tends to generate fewer distinct values, while a small step size tends to give a sample scattered not far from the initial value, thus failing to follow the target distribution. We choose 1000
as the number of proposed points for the Metropolis-Hastings algorithm.

To cover the areas with high gradient values, another random sample with size of 1000 is generated based on a multivariate normal distribution with mean at each support point. This normal distribution has the same covariance matrix as obtained from the current support points but divided by 10. When there is only one support point, the covariance matrix is chosen by the same strategy as described in the previous paragraph.

The support set expansion strategy of Wang (2008) is adopted here, which helps add new support points efficiently, as has been used in Chapter 3. If necessary, it doubles the support set at each iteration. Assuming that the current estimate of \( G \) consists of \( m \) support points, we first choose an arbitrary coordinate, and then the midpoints of every two neighboring support points on this coordinate axis as the break points. Hence the coordinate axis can be divided into \( m \) disjoint intervals \( C_1, \ldots, C_m \). Finally, the points with the largest, positive gradient value in each interval are added to the support set.

With a given \( \Theta \), the mixing proportions \( \pi \) are updated by using the second-order Taylor series expansion, which gives

\[
l(\pi, \Theta, \beta) - l(\pi', \Theta, \beta) \approx \frac{1}{2} \| S\pi' - 2 \|^2 - \frac{n}{2},
\]

where \( \Theta = (\theta_1, \ldots, \theta_m)^T \), \( s_i(\pi, \Theta, \beta) = \partial \log \{ f(y_i | X_i, Z_i, \pi, \Theta, \beta) \} / \partial \pi \), \( i = 1, \ldots, n \), \( S = (s_1(\pi, \Theta, \beta), \ldots, s_n(\pi, \Theta, \beta))^T \) and \( \| \cdot \| \) denotes the \( L_2 \)-norm. Therefore, by solving the following least square linear regression problem

\[
\min_{\pi'} \| S\pi' - 2 \|^2, \quad \text{subject to } \pi'^T 1 = 1, \pi' \geq 0,
\]

the likelihood \( l(\pi', \Theta, \beta) \) can thus be approximately maximized in the neighborhood of \( \pi \).
6.4.2 Updating \((G, \beta)\)

With a given number of support points, updating \((G, \beta)\) can be simply treated as fitting a finite mixture model. The quasi-Newton BFGS method is used here to solve an unconstrained optimization problem.

The BFGS method converges faster than the EM algorithm, for its superlinear order of convergence. It uses a sequence of matrices that approximate directly the inverse of the Hessian matrix, instead of computing the exact Hessian matrix and its inverse as in the Newton-Raphson method. Each approximating matrix is adjusted by a low-rank matrix at each iteration, with the negative identity matrix often chosen as the initial one. See Wang (2010) for some technical details about its use for updating \(G\) and \(\beta\) simultaneously.

For a discrete \(G\) with \(\pi\) and \(\Theta\), we list the first derivatives of \(l(G, \beta)\) with respect to \(\pi\), \(\Theta\) and \(\beta\) as follows:

\[
\frac{\partial l(G, \beta)}{\partial \pi} = \sum_{i=1}^{n} \left( f(y_i | X_i, Z_i, \theta_1, \beta), \ldots, f(y_i | X_i, Z_i, \theta_m, \beta) \right)^T / f(y_i | X_i, Z_i, G, \beta),
\]

\[
\frac{\partial l(G, \beta)}{\partial \Theta} = \sum_{i=1}^{n} \left( \frac{\partial f(y_i | X_i, Z_i, \theta_1, \beta)}{\partial \theta_1}, \ldots, \frac{\partial f(y_i | X_i, Z_i, \theta_m, \beta)}{\partial \theta_m} \right)^T / f(y_i | X_i, Z_i, G, \beta),
\]

\[
\frac{\partial l(G, \beta)}{\partial \beta} = \sum_{i=1}^{n} \left( \frac{\partial f(y_i | X_i, Z_i, G, \beta)}{\partial \beta} \right) / f(y_i | X_i, Z_i, G, \beta),
\]
where, for the binomial distribution with a logistic link and \(k = 1, \ldots, m\),

\[
\frac{\partial f(y_{i} | X_{i}, Z_{i}, \theta_{k}, \beta)}{\partial \theta_{k}} = \pi_{t+\frac{1}{2}} \prod_{j=1}^{n_{i}} p^{y_{ij}}_{ijk}(1 - p^{y_{i}}_{ijk})^{(n_{ij} - y_{ij})} \frac{y_{ij} - 2 y_{ij} p^{y_{i}}_{ijk} + p^{y_{i}}_{ijk} n_{ij}}{p^{y_{i}}_{ijk}(1 - p^{y_{i}}_{ijk})} \frac{\partial p^{y_{i}}_{ijk}}{\partial \theta_{k}},
\]

\[
\frac{\partial p^{y_{i}}_{ijk}}{\partial \theta_{k}} = \frac{x_{ij}}{\left\{1 + \exp(x_{ij}^\top \theta_{k} + z_{ij}^\top \beta)\right\}^2}.
\]

\[
\frac{\partial f(y_{i} | X_{i}, Z_{i}, G, \beta)}{\partial \beta} = \sum_{k=1}^{m} \pi_{t} \prod_{j=1}^{n_{i}} p^{y_{ij}}_{ijk}(1 - p^{y_{i}}_{ijk})^{(n_{ij} - y_{ij})} \frac{y_{ij} - 2 y_{ij} p^{y_{i}}_{ijk} + p^{y_{i}}_{ijk} n_{ij}}{p^{y_{i}}_{ijk}(1 - p^{y_{i}}_{ijk})} \frac{\partial p^{y_{i}}_{ijk}}{\partial \beta},
\]

\[
\frac{\partial p^{y_{i}}_{ijk}}{\partial \beta} = \frac{z_{ij}}{\left\{1 + \exp(x_{ij}^\top \theta_{k} + z_{ij}^\top \beta)\right\}^2}.
\]

We summarize the above in the algorithm of the CNMM-MS for computing a MLE \((G, \beta)\). The methods for generating random grids in the CNMM algorithm here differ from the way described in Section 3.3.3.

**Algorithm 5 (CNMM-MS)**. Set \(t = 0\). From an initial \(G_{0}\) with finite support, an initial \(\beta_{0}\) and \(l_{h}(G_{0}, \beta_{0}) < -\infty\), repeat the following steps.

**Step 1**: Generate a random sample \(\theta_{1}^{*}, \ldots, \theta_{n_{g}}^{*}\) consisting of two random grids.

The first random grid is generated by using the MCMC method from density \(d^{*}(\theta; G_{t}, \beta_{t})\) as defined by (6.10), and the second random grid is generated from the current support points. Then divide them into \(m + 1\) groups, based on the \(m\) intervals on an arbitrarily chosen coordinate which are defined by the current support points.

**Step 2**: Let \(\Theta^{+}\) contain \(\{\theta_{l}^{*} : d_{H}(\theta_{l}^{*}; G_{t+\frac{1}{2}}) > 0, l = 1, \ldots, m\}\). Set \(\Theta_{t+\frac{1}{2}} = (\Theta_{t}, \Theta^{+})\) and \(\pi_{t+\frac{1}{2}} = (\pi_{t}^T, 0^T)^T\). Find \(\pi_{t+\frac{1}{2}}\), by solving problem (6.11), followed with a line search.

**Step 3**: Discard all support points with zero entries in \(\pi_{t+\frac{1}{2}}\), which gives \(G_{t+\frac{1}{2}}^{+}\) with \(\Theta_{t+\frac{1}{2}}^{+}\) and \(\pi_{t+\frac{1}{2}}^{+}\).
6.5 Demonstration of Algorithm 5

In this section, two demonstrations are provided to show the process of finding an NPMLE with a given \( \beta \) and the process of finding a joint MLE \((\hat{G}, \hat{\beta})\). We considered a binomial model with the logistic link function,

\[
    f(y_i | X_i, Z_i, \theta_i, \beta) = \prod_{j=1}^{n_i} \left\{ q(x_{ij}^\top \theta_i + z_{ij}^\top \beta)^{y_{ij}} (1 - q(x_{ij}^\top \theta_i + z_{ij}^\top \beta))^{n_{ij} - y_{ij}} \right\},
\]

where \( \text{logit}(\mu_{ij}) = x_{ij}^\top \theta_i + z_{ij}^\top \beta, \mu_{ij} = E[y_{ij} | x_{ij}, z_{ij}, \theta_i, \beta] \) and \( q(\mu_{ij}) = \frac{\exp(\mu_{ij})}{1 + \exp(\mu_{ij})} \).

Here \( \theta \) contains the random effects and \( \beta \) the fixed effects. A data set is generated from a semiparametric mixture model with \( \theta \) following a discrete multivariate mixture with mass 0.5 on each of the two support pints, \((2, 2)^T\) and \((1, 1)^T\) and \( \beta = (0, 3)^T \). It contains 100 groups, each of size \( n_i \in \{6, 7, 8, 9, 10\} \) with equal proportions. Apart from the slope, each covariate is an independent normal deviate with mean \(-0.5\) and unit variance.

The first demonstration uses the variant of the CNMM-MS to find an NPMLE with a given \( \beta \). A random grid of 200 points, 100 from the MCMC method and 100 around current support points, is used. Since \( \beta \) is given, the BFGS is only used to update \( \pi \) and \( \Theta \) by finding the exact locations for support points. We set

**Step 4:** Run the full BFGS algorithm, which updates \( G_{t+\frac{1}{2}}^+ \) to \( G_{t+1} \) and \( \beta_t \) to \( \beta_{t+1} \).

**Step 5:** If \( I_H(G_{t+1}, \beta_{t+1}) - I_H(G_t, \beta_t) \leq \tau \) (a small positive threshold value), break; otherwise set \( t = t + 1 \).

An initial \( G_0 \) and an initial \( \beta_0 \), can be chosen from the estimates by fitting the corresponding generalized linear model, regardless of group information. Moreover, when \( \beta \) is fixed, the CNMM-MS algorithm, as a variant, reduces to compute the NPMLE \( \hat{G} \) only.
Figure 6.1: 3 iterations of the CNMM for finding a NPMLE $\hat{G}$. Each plot represents contours of the gradient function. Random points generating from the MCMC are in grey and “plus”, random points generating around current support points in grey and “circle”, current support points in red and “star” and the new support points are in red and “circle”.
\( \beta = (0, 3)^T \), and the CNMM used 4 iterations to find the \( \hat{G} \). 3 plots are shown in Figure 6.1 represent results after step 2 of the CNMM-MS in the 1st, 2nd and 4th iterations for finding \( \hat{G} \) for the gradient function.

It can be seen that, in this example, the MCMC is able to generate dispersed random points to cover areas with big gradient values and the random samples which are generated from multivariate normal distributions provide a good coverage in the local area around each supper point. With such random grids, the algorithm is able to locate reasonably well all the local maxima of the gradient function at each iteration. The CNMM took 4 iterations to converge. \( \hat{G} \) has 5 support points and a log-likelihood value of \(-272.11\). Table 6.5 gives the support points and their associated masses given by \( \hat{G} \).

Table 6.1: Computing \( \hat{G} \) using the CNMM for the simulated sample with a given \( \beta \).

<table>
<thead>
<tr>
<th>( \theta_j )</th>
<th>( \pi_j )</th>
<th>log-likelihood</th>
<th>max gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1.00, 4.07)</td>
<td>0.03</td>
<td>-272.11</td>
<td>2.06 \times 10^{-7}</td>
</tr>
<tr>
<td>(0.98, 1.26)</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.09, 0.88)</td>
<td>0.66</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.50, 1.40)</td>
<td>0.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.47, 2.74)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The second demonstration is about finding the joint MLE \( (\hat{G}, \hat{\beta}) \) by the CNMM-MS. We used the same simulated data set and the algorithm took 4 iterations to converge. Figure 6.2 shows three selected iterations of finding \( (\hat{G}, \hat{\beta}) \). It contains three plots and each of them is plotted after step 2 of the CNMM-MS. Table 6.2 gives the values of \( (\hat{G}, \hat{\beta}) \), where \( \hat{G} \) consists of 5 components and the log-likelihood value of \( (\hat{G}, \hat{\beta}) \) is \(-272.08\).

### 6.6 The Direct Search Directional Derivative Method

Lesperance et al. (2014) has also applied semiparametric mixtures to estimating GLMM’s. Since their algorithm also uses the gradient function to help find the
Table 6.2: \((\hat{G}, \hat{\beta})\) from the CNMM-MS for the simulated sample.

<table>
<thead>
<tr>
<th>(\theta_j^T)</th>
<th>(\hat{\pi}_j)</th>
<th>(\hat{\beta}^T)</th>
<th>log-likelihood</th>
<th>max gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1.02, 4.04)</td>
<td>0.03</td>
<td>-0.02, 3.00</td>
<td>-272.08</td>
<td>1.14 \times 10^{-6}</td>
</tr>
<tr>
<td>(0.98, 1.25)</td>
<td>0.28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.07, 0.92)</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.50, 1.40)</td>
<td>0.61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.47, 2.74)</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NPMLE, at each iteration it needs to find the local maxima of the gradient function and update the support point set by including these local maxima. They proposed the Direct Search Directional Derivative (DSDD) method to compute an NPMLE. At each iteration, the DSDD firstly updates the support point set and then the mixing proportions.

The DSDD uses the coordinate search method, a subclass of direct search methods, to find the local maxima of the gradient function and include them in the support point set. A constant generating matrix \(C\) containing all the possible pairs of \(\{-1, 0, 1\}\) is used to provide all the potential candidates for gradient evaluation. It is given by \(C_s = \delta_s \ast C_{s-1} + \theta_s\), where \(s\) is the current iteration, \(\delta_s\) the step size for iteration \(s\), \(\theta_s\) the support point \(\theta\) at \(s\) iteration. The initial \(C_0\) for a two dimensional space is given by

\[
C_0 = \begin{bmatrix}
1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\
-0 & 1 & 0 & -1 & 1 & -1 & -1 & 1 & 0
\end{bmatrix}.
\]

Here each column of \(C\) represents a potential candidate of support points. In details, for each current support point, it searches \(3^d - 1\) points to find the direction which increases the gradient function mostly, where \(d\) indicates the dimensionality. The \(3^d - 1\) checking points represent different coordinates from the current support point and a step size is used here to control the distance from the checking points to the current support point.

After finding the local maxima of the gradient function and including them into
Figure 6.2: 3 iterations of the CNMM-MS for finding $(\hat{G}, \hat{\beta})$. Each plot represents contours of the gradient function. Random points generating from the MCMC are in grey and “plus”, random points generating around current support points in grey and “circle”, current support points in red and “star” and the new support points are in red and “circle”.

$0th$ iterate $\beta_1 = (-0.02, 2.92)^T$

$1st$ iterate $\beta_2 = (-0.01, 2.97)^T$

$4th$ iterate $\hat{\beta} = (-0.02, 3.00)^T$
the support points set, the DSDD makes use of the CNM algorithm (Wang, 2007) to update the mixing proportions.

To compute a joint MLE (\( \hat{G}, \hat{\beta} \)), Lesperance et al. (2014) proposed two algorithms, the DSDD-AP and the DSDD-PL, by making use of the CNM-AP and the CNM-PL (Wang, 2010). For the DSDD-AP, they firstly use the DSDD algorithm to maximize \( G \) and then use the BFGS algorithm to maximize the likelihood function of \( \beta \) and repeat these two steps until converge. For the DSDD-PL, the difference is that \( \beta \) is estimated by maximizing the profile likelihood \( \tilde{l}(\beta) = l(\beta, \hat{G}_\beta) = \max_G l(\beta, G) \).

According to their experimental results, these two methods are able to produce very similar results for fixed effects as well as the log-likelihood values. Moreover, the DSDD-AP is faster.

Two demonstrations of their algorithm are given below. Since there is no published code for the methods of Lesperance et al. (2014), I implemented them based on the description in their paper. As reported by Lesperance et al. (2014), the DSDD-AP and DSDD-PL estimates were very similar, I only reported the results of the DSDD algorithm and the DSDD-AP algorithm. We still use the same simulated data set as described in Section 6.5. With a given \( \beta = (0, 3)^T \), the DSDD took 16 iterations to find \( \hat{G} \). Figure 6.3 shows the results of three iterations after finding the local maxima of the gradient function by using the coordinate search method. Each plot shows the contour lines of gradient function, with the current support points in red and “star” and the new support points in red and “circle”. \( \hat{G} \) has five support points, with a log-likelihood value of \(-272.11\), as given in Table 6.3.

Table 6.3: Estimates from the DSDD for the simulated sample with a given \( \beta \).

<table>
<thead>
<tr>
<th>( \theta_j^k )</th>
<th>( \hat{\pi}_j )</th>
<th>log-likelihood</th>
<th>max gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1.00, 4.07)</td>
<td>0.03</td>
<td>(-272.11)</td>
<td>(1.76 \times 10^{-04})</td>
</tr>
<tr>
<td>(0.98, 1.26)</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.09, 0.88)</td>
<td>0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.50, 1.40)</td>
<td>0.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.47, 2.74)</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 6.3: 3 iterations of the DSDD for finding a NPMLE \( \hat{G} \). Each plot represents contour lines of gradient function with current support points in red and “star” and the new support points are in red and “circle”.

Figure 6.4 shows the process of finding a joint MLE for \((G, \beta)\) by using the DSDD-AP. The DSDD-AP took 6 iterations to converge. \( \hat{G} \) also has five support points as shown in Table 6.4, and is very similar to that given by the CNMM-MS.
6.7 Difference between two methods for fitting semiparametric mixtures

In this section, differences between our proposed method and the methods of Lesperance et al. (2014) are studied.
6.7. DIFFERENCE BETWEEN TWO METHODS FOR FITTING SEMIPARAMETRIC MIXTURES

Table 6.4: \((\hat{G}, \hat{\beta})\) from the DSDD-AP for the simulated sample.

<table>
<thead>
<tr>
<th>( \theta_j^* )</th>
<th>( \hat{\pi}_j )</th>
<th>( \hat{\beta}_j^* )</th>
<th>log-likelihood</th>
<th>max gradient</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-1.02, 4.04)</td>
<td>0.03</td>
<td>-0.02, 3.00</td>
<td>-272.08</td>
<td>1.14 \times 10^{-6}</td>
</tr>
<tr>
<td>(0.98, 1.25)</td>
<td>0.28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.07, 0.92)</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1.50, 1.40)</td>
<td>0.61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2.47, 2.74)</td>
<td>0.03</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Firstly, the methods of Lesperance et al. (2014) do not guarantee to find the NPMLE. Because the gradient function characterizes the NPMLE, so the key to the NPMLE is the ability to find the global maxima of the gradient function. Lesperance et al. (2014) used a local search method for locating local maxima of the gradient function which may fail, in principle, to locate global maximum. In contrast, the proposed method used a grid search to find all the local maxima of the gradient function and with a sufficiently fine grid, it can always find the NPMLE.

Secondly, the two methods differ in the computational complexity for finding the NPMLE. Lesperance, et al., (2014) used a direct search method to update the support points. For one updating of the support points set, at each iteration it examined \(3^d - 1\) points for each support point to find a point where the gradient value will increase. Here \(d\) indicates the dimensionality, and hence the computational complexity increases exponentially with dimensions. Our proposed method uses a random grid for a crude search for the local maxima of the gradient function and the BFGS is used to compute the exact locations of the support points. Since the BFGS can eventually find the exact locations of the support points, a random grid only needs to locate their rough areas. As a result, size of a random grid does not need to increase with dimensions. Therefore, the computational complexity of the proposed method is very much independent of dimensionality.

Lastly, both two methods for computing a semiparametric MLE are developed based on the algorithm proposed by Wang (2010). Wang (2010) described three algorithms for computing a semiparametric MLE. The first algorithm, called CNM-
AP, alternatively uses one iteration of the CNM to update $G$ with $\beta$ fixed and the BFGS method to update $\beta$ with $G$ fixed. The second algorithm, called CNM-PL, maximizes the profile likelihood function of $\beta$ by the BFGS where any profile likelihood value at a fixed point of $\beta$ is provided by the CNM. The third algorithm, called CNM-MS, alternatively performs one iteration of the CNM and updates all parameters including $G$ and $\beta$ by the BFGS. According to Wang’s (2010) results, the performance of the CNM-AP algorithm is consistently good but the efficiency of the CNM-AP algorithm depends on the correlation between $G$ and $\beta$. The CNM-PL algorithm is generally more time-consuming than the CNM-MS. As pointed out by Wang (2010), the CNM-MS has the best performance and it is also the most difficult to implement. The methods of Lesperance, et al., (2014) are based on first two algorithms, namely CNM-PL and CNM-AP, while our proposed method is based on the third algorithm, CNM-MS.

Because the CNMM-MS and DSDD-AP are used for solving the same maximum likelihood estimation problem, so their estimates are, not surprisingly, very similar, as given in Section 6.5 and 6.6. However, they differ in computational times taken to find an NPMLE. A time comparison of the two was made using the above same simulated data set. It can be seen from Table 6.5 that the CNMM-MS is much faster than the DSDD-AP. There are two reasons. One is that the CNMM took less iterations for finding all the local maxima of the gradient function and for updating the support point set by using random grids. The other is that the CNMM-MS is based on the CNM-MS, which should be faster than the CNM-AP, according to Wang (2010).

<table>
<thead>
<tr>
<th>Method</th>
<th>CNMM/DSDD (s)</th>
<th>BFGS (s)</th>
<th>Total time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMM-MS</td>
<td>14.58</td>
<td>6.01</td>
<td>21.19</td>
</tr>
<tr>
<td>DSDD-AP</td>
<td>305.44</td>
<td>3.67</td>
<td>309.04</td>
</tr>
</tbody>
</table>

Table 6.5: Time comparison between the CNMM-MS and the DSDD-AP for computing a joint MLE of $(G, \beta)$. 
6.8 Summary

In this chapter, we studied GLMM’s with multiple random effects by using semi-parametric mixtures. In particular, we make no assumption about the distribution of the random effects and use the non- and semi-parametric maximum likelihood estimation method to estimate the distribution. An algorithm is developed based on the CNMNB and the BFGS algorithm. The proposed method works in general for all families of GLMM’s, although we take binomial distributions with the logistic link function as a case study in our investigation. We also investigated the method of Lesperance et al. (2014), and made comparisons between the two algorithms.

Demonstrations of both the proposed method and the DSDD-AP (Lesperance et al., 2014) are provided. These two methods produce similar estimates, but our proposed method is much faster than the DSDD-AP.
Chapter 7

Numerical Studies of Generalized Linear Mixed-Effects Models

7.1 Introduction

In this chapter, we study several simulated and two real-world data sets. Simulated studies show that this semiparametric formulation outperforms significantly the parametric method that is under the normality assumption, in terms of both estimating the random and fixed effects, especially in situations when covariates are highly correlated and random effects follow a non-normal distribution. Real-world data demonstrates that without using normality assumption the new method gives similar estimates of the fixed effects to the traditional method, yet with greater likelihood values.

7.2 Simulation Studies

The focus of the simulation studies reported below is on how the accuracy of estimated fixed effects is affected by the distribution of random effects and the correlation level among covariates. Therefore, the comparisons among different methods only focused on the fixed effects.
CHAPTER 7. NUMERICAL STUDIES OF GENERALIZED LINEAR MIXED-EFFECTS MODELS

The nonparametric mixture-based method and the parametric method under the normality assumption in the R package \texttt{lmee} (lme4) (Bates et al., 2014a) were compared. Because the CNMM-MS and the DSDD-AP (Lesperance et al., 2014) are supposed to find the MLEs of the same models, here we only report the results of the CNMM-MS for its higher computation efficiency and accuracy as demonstration in Chapter 6.

In this chapter, we considered a binomial model with the logistic link function,

\[
f(y_i | X_i, Z_i, \theta_i, \beta) = \prod_{j=1}^{n_i} \left\{ q(x_{ij}^\top \theta_i + z_{ij}^\top \beta) y_{ij} \left[ 1 - q(x_{ij}^\top \theta_i + z_{ij}^\top \beta) \right]^{n_i - y_{ij}}, \right.\]

where \( \text{logit}(\mu_{ij}) = x_{ij}^\top \theta_i + z_{ij}^\top \beta, \mu_{ij} = E[y_{ij} | x_{ij}, z_{ij}, \theta_i, \beta] \) and \( q(\mu_{ij}) = \frac{\exp(\mu_{ij})}{1 + \exp(\mu_{ij})} \).

Here \( \theta \) are random effects, \( \beta \) fixed effects, \( z_{ij} \) covariates of random effects and \( x_{ij} \) covariates of fixed effects.

### 7.2.1 Bivariate Random Effects

#### Setup

We considered the situation with two random effects and two fixed effects. The resulting mixed model is

\[
\text{logit}(\mu_{ij}) = \theta_{0i} + \theta_{1i} z_{1ij} + \beta_1 z_{2ij} + \beta_2 z_{3ij}, \tag{7.1}
\]

where \( \theta_i = (\theta_{0i}, \theta_{1i})^T \) denotes the random effects which are assumed to vary over groups, and \( \beta = (\beta_1, \beta_2)^T \) the fixed effects. \( z_{1ij} \) is a covariate of random effects and \( (z_{2ij}, z_{3ij}) \) are covariates of fixed effects. Because \texttt{lmee} assumes that the distribution of random effects follows a multivariate normal distribution with mean zero, the fixed effects \((\beta_{0i}, \beta_{1i})\) for \((\theta_{0i}, \theta_{1i})\) are also included in the model which is used by \texttt{lmee}. Since the CNMM-MS uses semiparametric mixture models without assuming normality assumption, the mean of \( G(\theta) \) directly provides the estimates of \((\beta_{0i}, \beta_{1i})\).
To study the influence of the distribution of random effects, random effects $\theta_i = (\theta_0i, \theta_1i)^T$ were generated from five bivariate distributions, as labelled from A to E in Table 7.1. Distributions A and B are discrete and supported at two points with different distances between them. Distribution C is a multivariate normal. Distributions D and E are multivariate skew-normal with skewness controlled by shape parameters $\alpha$ (Azzalini and Valle, 1996). The density function of the skew-normal is

$$f(x) = 2\phi(x; \Sigma)\Phi(\alpha^T x),$$

where $\phi(x; \Sigma)$ is a multivariate normal density with mean 0 and covariance matrix $\Sigma$, and $\Phi(\cdot)$ is the univariate standard normal distribution function. Contour plots of their density functions are shown in Fig. 7.2.1. Table 7.2 gives the values of fixed effects $(\beta_{\theta_0}, \beta_{\theta_1}, \beta_1, \beta_2)$ for the five simulated models, where $(\beta_{\theta_0}, \beta_{\theta_1})$ is the mean of each distribution in Table 7.1, and the true values for $(\beta_1, \beta_2)$ are (0, 3).

To study the influence of the correlation level among covariates of $(z_{1ij}, z_{2ij}, z_{3ij})$,
\[ \alpha = (0, 0)^T \quad \alpha = (0, 4)^T \quad \alpha = (0, 8)^T \]

Figure 7.1: Contour plots for densities C, D and E given in Table 7.1.

Firstly we define the covariance matrix of covariates as

\[
\Omega(\rho) = \begin{pmatrix}
1 & \rho & \rho \\
\rho & 1 & \rho \\
\rho & \rho & 1
\end{pmatrix},
\]

where \( \rho \) is the correlation coefficient between a pair of covariates. The covariates of \((z_{1ij}, z_{2ij}, z_{3ij})\) were drawn from \(N_3(0, \Omega(\rho))\). Here \(N_d(0, \Sigma)\) denotes a \(d\)-variate normal distribution with mean \(0\) and covariance matrix \(\Sigma\).

### Results

In this section, we studied three situations of \(\rho = 0, 0.5\) and \(0.9\). For each value of \(\rho\) and each distribution from Table 7.1, we generated 1000 data sets. Every data set contains 100 groups, each of size \(n_i \in \{6, 7, 8, 9, 10\}\) with equal proportions. Therefore, we have 15 different combinations of \(\rho\) and the underlying distribution of random effects.

Tables 7.3 – 7.5 provide a summary of the simulation results. To evaluate the performance, the mean square errors of \(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3\) were calculated, from 1000 repeated samplings along with their standard errors given in parentheses. Paired \(t\)-tests were also conducted to compare square errors, with the resulting \(p\)-values given...
in the table. A mean square error that is in boldface in the table is the smaller one between the two competitors, when the paired $t$-test gives a $p$-value $<0.1$.

Simulation results for the five distributions of random effects, when the correlation between the covariates is $\rho = 0$, are given in Table 7.3. It can be seen that when the covariates are generated independently, the performance of the lme4 and that of the CNMM-MS have no significant difference in most cases. Only for the discrete distribution B where two mass points have a bigger distance, the CNMM-MS performs better than the lme4 at estimating $\beta_1$.

<table>
<thead>
<tr>
<th>Estimator (A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(D)</th>
<th>(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_{\theta_0}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>2.02 (0.26)</td>
<td>3.12 (0.41)</td>
<td>0.72 (0.10)</td>
<td>0.39 (0.12)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>2.19 (0.20)</td>
<td>3.44 (0.36)</td>
<td>0.65 (0.12)</td>
<td>0.42 (0.08)</td>
</tr>
<tr>
<td>($p$-value)</td>
<td>0.8331</td>
<td>0.3218</td>
<td>0.4646</td>
<td>0.7913</td>
</tr>
<tr>
<td>$\beta_{\theta_1}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>1.02 (0.11)</td>
<td>2.93 (0.55)</td>
<td>0.45 (0.12)</td>
<td>2.46 (0.45)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>0.97 (0.15)</td>
<td>3.27 (0.71)</td>
<td>0.53 (0.15)</td>
<td>2.21 (0.37)</td>
</tr>
<tr>
<td>($p$-value)</td>
<td>0.8245</td>
<td>0.4672</td>
<td>0.6336</td>
<td>0.6847</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>0.53 (0.08)</td>
<td>1.18 (0.16)</td>
<td>0.99 (0.11)</td>
<td>0.73 (0.14)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>0.54 (0.06)</td>
<td>0.81 (0.09)</td>
<td>1.03 (0.11)</td>
<td>0.77 (0.11)</td>
</tr>
<tr>
<td>($p$-value)</td>
<td>0.8868</td>
<td>0.0468</td>
<td>0.5881</td>
<td>0.6554</td>
</tr>
<tr>
<td>$\beta_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>2.19 (0.25)</td>
<td>4.79 (0.64)</td>
<td>3.61 (0.49)</td>
<td>3.55 (0.56)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>2.25 (0.31)</td>
<td>6.24 (0.95)</td>
<td>4.03 (0.45)</td>
<td>3.74 (0.43)</td>
</tr>
<tr>
<td>($p$-value)</td>
<td>0.6875</td>
<td>0.1475</td>
<td>0.2681</td>
<td>0.5743</td>
</tr>
</tbody>
</table>

Table 7.3: Simulation results for 5 target distributions, when $\rho = 0$ (Mean Square Error $\times 10^2$).

Table 7.4 gives simulation results for the five distributions of random effects, when $\rho = 0.5$. The CNMM-MS performs slightly better than the lme4 in all cases. However, the difference between the performance of the lme4 and CNMM-MS is clear but perhaps not remarkable yet.

Table 7.5 provides simulation results for the five distributions of random effects, when $\rho = 0.9$. It is easy to see that the CNMM-MS performs better than the lme4 for distribution B, D and E at estimating all the fixed effects. For distribution A,
### Chapter 7. Numerical Studies of Generalized Linear Mixed-Effects Models

#### Table 7.4: Simulation results for 5 target distributions and correlation of covariates $\rho = 0.5$ (Mean Square Error × 10²).

where the two mass points of the discrete distribution have a smaller distance, the performance of these two methods are comparable. The CNMM-MS and the `lme4` have similar performance when random effects follow a normal distribution.

#### Table 7.5: Simulation results for 5 target distributions, when $\rho = 0.9$. (Mean Square Error × 10²).

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(D)</th>
<th>(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho = 0.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta_{\theta_0}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>lme4</code></td>
<td>2.21 (0.36)</td>
<td>3.33 (0.21)</td>
<td>1.32 (0.20)</td>
<td>3.26 (0.38)</td>
<td>4.46 (0.32)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>2.19 (0.30)</td>
<td>3.49 (0.36)</td>
<td>1.67 (0.22)</td>
<td>3.13 (0.29)</td>
<td>4.11 (0.28)</td>
</tr>
<tr>
<td>$p$-value</td>
<td>0.7573</td>
<td>0.6192</td>
<td>0.2240</td>
<td>0.3210</td>
<td>0.1142</td>
</tr>
<tr>
<td>$\rho = 0.9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta_{\theta_1}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>lme4</code></td>
<td>1.08 (0.14)</td>
<td>1.89 (0.28)</td>
<td>1.20 (0.15)</td>
<td>1.70 (0.25)</td>
<td>1.78 (0.27)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>1.15 (0.16)</td>
<td><strong>1.39</strong> (0.18)</td>
<td>1.32 (0.16)</td>
<td>1.71 (0.22)</td>
<td>1.45 (0.15)</td>
</tr>
<tr>
<td>$p$-value</td>
<td>0.3371</td>
<td>0.0265</td>
<td>0.3129</td>
<td>0.9615</td>
<td>0.1814</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>lme4</code></td>
<td><strong>3.53</strong> (0.45)</td>
<td>4.91 (0.68)</td>
<td>3.12 (0.55)</td>
<td>3.52 (0.53)</td>
<td>3.48 (0.57)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>4.05 (0.54)</td>
<td>4.58 (0.57)</td>
<td>3.25 (0.46)</td>
<td>3.50 (0.71)</td>
<td><strong>2.87</strong> (0.35)</td>
</tr>
<tr>
<td><code>lme4</code></td>
<td>3.89 (0.51)</td>
<td>11.81 (0.78)</td>
<td>7.71 (0.54)</td>
<td>7.41 (0.54)</td>
<td>6.86 (0.42)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>3.92 (0.46)</td>
<td><strong>7.11</strong> (0.67)</td>
<td>7.09 (0.62)</td>
<td><strong>5.36</strong> (0.55)</td>
<td><strong>5.18</strong> (0.25)</td>
</tr>
</tbody>
</table>

Table 7.4: Simulation results for 5 target distributions and correlation of covariates $\rho = 0.5$ (Mean Square Error × 10²).

Table 7.5: Simulation results for 5 target distributions, when $\rho = 0.9$. (Mean Square Error × 10²).
In summary, it is clear that the performance of an estimator is affected by both the distribution of random effects and the correlation level among covariates. For example, for distribution A, the discrete distribution of random effects where two mass points are reasonably close to each other, there is no significant difference between the performance of the CNMM-MS and \texttt{lme4}, regardless of the values of \(\rho\). For distribution B, where two mass points are well separately, the CNMM-MS performs much better than \texttt{lme4}, especially when \(\rho = 0\) where the covariates come from a highly correlated distribution. However, there is hardly any significant difference between the performance of these two methods for \(\rho = 0\) and \(\rho = 0.5\). For distribution C, no significant difference between the performance of these two methods can be seen, for all \(\rho\)-values. For the skew-normal distribution D, the outperformance of the CNMM-MS is only significant when \(\rho = 0.9\). For the strongly skewed normal distribution E, the CNMMS-MB dominates the \texttt{lme4} when \(\rho = 0.5\) and 0.9. In conclusion, the CNMM-MS performs better when covariates are correlated and when random effects follow a severely non-normal distribution, and similarly, if otherwise.

These findings show that the estimation of the fixed effects is affected if the distribution of random effects is misspecified, especially when the covariates are correlated. Neuhaus et al. (1992) suggested that the normality assumption of the distribution of random effects can be robust for estimating the fixed effects. They only studied the random intercept situation under five different distributions, i.e., gamma distributions with scale parameter 1 and shape parameters 0.5 and 16, \(t\)-distributions with 3 and 5 degrees of freedom, and a normal distribution, where the covariates were independently generated from a standard normal distribution. Heagerty and Kurland (2001) used similar designs to those of Neuhaus et al. (1992), but explored a wider range of values for the shape and the scale parameter in gamma distribution. They concluded that substantial bias in the MLEs for the fixed effects can be caused by the misspecification of distribution of the random
intercept as the gamma shape parameter increases. Agresti et al. (2004) studied four distributions of the random intercept, normal, two-point discrete, uniform and exponential distribution, and showed that the MLEs of the fixed effects are very sensitive to the assumption of the distribution of random effects. However, their studies only considered one random effect which is the intercept and did not touch up correlated covariates. Our findings shed a new light on the effects of a misspecified distribution of random effects where there are multiple random effects and correlated covariates.

### 7.2.2 Multivariate Random Effects

#### Setup

We considered the situation with five random effects and two fixed effects. Similarly to the bivariate case, random effects $\mathbf{\theta}_i = (\theta_0i, \theta_1i, \theta_2i, \theta_3i, \theta_4i)^T$ are assumed to vary over groups, and $\mathbf{\beta} = (\beta_1, \beta_2)^T$ are the fixed effects. $(z_{1ij}, z_{2ij}, z_{3ij}, z_{4ij})$ are the covariates of random effects and $(z_{5ij}, z_{6ij})$ are the covariates of fixed effects. It gives us a mixed model,

$$\text{logit}(\mu_{ij}) = \theta_{0i} + \theta_{1i}z_{1ij} + \theta_{2i}z_{2ij} + \theta_{3i}z_{3ij} + \theta_{4i}z_{4ij} + \beta_1z_{5ij} + \beta_2z_{6ij}.$$  \(7.2\)

Similarly, `lme4` assumes that the distribution of random effects follows a multivariate normal distribution with mean zero, so the fixed effects $(\beta_{0i}, \beta_{1i}, \beta_{2i}, \beta_{3i}, \beta_{4i})$ for $(\theta_{0i}, \theta_{1i}, \theta_{2i}, \theta_{3i}, \theta_{4i})$ are also included in the model which is used by `lme4`. Also, the mean of $G(\mathbf{\theta})$ directly provides the estimates of $(\beta_{0i}, \beta_{1i}, \beta_{2i}, \beta_{3i}, \beta_{4i})$.

We studied five 5-dimensional distributions from which random effects $(\theta_{0i}, \theta_{1i}, \theta_{2i}, \theta_{3i}, \theta_{4i})$ were generated, as labelled from F to J in Table 7.6. Distributions F and G are discrete and supported at two points with different distances between them. Distribution H is a multivariate normal. Distributions I and G are multivariate skew-normal with skewness controlled by shape parameters $\alpha$. Table 7.7 gives the values of fixed
effects \((\beta_{\theta_0}, \beta_{\theta_1}, \beta_{\theta_2}, \beta_{\theta_3}, \beta_{\theta_4}, \beta_1, \beta_2)\) for the five simulated models, where \((\beta_{\theta_0}, \beta_{\theta_1}, \beta_{\theta_2}, \beta_{\theta_3}, \beta_{\theta_4})\) is the mean of each distribution in Table 7.6, and the true values for \((\beta_1, \beta_2)\) are \((0, 3)\). We considered two different correlation levels among covariates. The covariates of \((z_{1ij}, z_{2ij}, z_{3ij}, z_{4ij}, z_{5ij}, z_{6ij})\) were drawn from \(N_6(0, \Omega(\rho))\) where \(\rho = 0\) and \(\rho = 0.9\), respectively.

Since it is difficult to provide contour plots for densities in high dimensional space, to demonstrate the distribution I and J, we generate 500 observations for each density, as shown in the pairwise scatter plots in Fig. 7.2 and Fig 7.3, respectively.

Figure 7.2: Pairwise scatter plots for 500 observations generated from distribution I in Table 7.6.
CHAPTER 7. NUMERICAL STUDIES OF GENERALIZED LINEAR MIXED-EFFECTS MODELS

Figure 7.3: Pairwise scatter plots for 500 observations generated from distribution J in Table 7.6.

Results

We investigated two situations, $\rho = 0$ and $0.9$, respectively. Similarly, we generated 1000 data sets for each value of $\rho$ and each distribution from Table 7.6. There are 100 groups in each data set contains 100 groups, each of size $n_i \in \{6, 7, 8, 9, 10\}$ with equal proportions and hence it gives 10 different combinations of $\rho$ and the underlying distribution of random effects.

Performance was evaluated by the mean square errors of $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4,$
7.2. SIMULATION STUDIES

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F) two-point discrete distribution</td>
<td>( \theta_1 = (0, 0, 0, 0)^T, \theta_2 = (2, 2, 2, 2)^T, \pi = (0.5, 0.5)^T )</td>
</tr>
<tr>
<td>(G) two-point discrete distribution</td>
<td>( \theta_1 = (0, 0, 0, 0)^T, \theta_2 = (4, 4, 4, 4)^T, \pi = (0.5, 0.5)^T )</td>
</tr>
<tr>
<td>(H) normal distribution</td>
<td>( \mu = (0, 0, 0, 0)^T, \Sigma ) is a unit matrix of size 5, ( \alpha = (4, 4, 4, 4)^T )</td>
</tr>
<tr>
<td>(I) skew-normal distribution</td>
<td>( \mu = (0, 0, 0, 0)^T, \Sigma ) is a unit matrix of size 5, ( \alpha = (8, 8, 8, 8)^T )</td>
</tr>
</tbody>
</table>

Table 7.6: 5 5-dimensional distributions of random effects used in the simulation studies.

<table>
<thead>
<tr>
<th>Name</th>
<th>( \beta_{\theta_1} )</th>
<th>( \beta_{\theta_2} )</th>
<th>( \beta_{\theta_3} )</th>
<th>( \beta_{\theta_4} )</th>
<th>( \beta_1 )</th>
<th>( \beta_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>(G)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>(H)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>(I)</td>
<td>0.35</td>
<td>0.35</td>
<td>0.35</td>
<td>0.35</td>
<td>0.35</td>
<td>0</td>
</tr>
<tr>
<td>(J)</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
<td>0.36</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.7: Fixed effects for 5 distributions in the simulation studies.

\( \hat{\beta}_1 \) and \( \hat{\beta}_2 \), from 1000 random samples along with their standard errors given in parentheses. We conducted paired \( t \)-tests to compare square errors, and provided the resulting \( p \)-values in the table. A mean square error that is in boldface in the table is the smaller one between the two competitors, when the paired \( t \)-test gives a \( p \)-value \(< 0.1 \).

Table 7.8 provides simulation results for the five distributions of random effects, when \( \rho = 0 \). It is clear that the CNMM-MS significantly outperforms the \texttt{lme4} for distribution G, where two mass points of the discrete distribution are far away from each other. The performance of these two methods are not significantly different for the other distributions. The results are quite similar to the cases of two random effects.

Simulation results for the five distributions of random effects, when \( \rho = 0.9 \) are provided in Table 7.9. It can be seen that the CNMM-MS performs significantly better than the \texttt{lme4} for distributions F, G and J, where covariates are highly correlated. Moreover, the CNMM-MS and the \texttt{lme4} have similar performance for distributions H and J. For distribution H, the \texttt{lme4} is supposed to have similar performance to that of the CNMM-MS since random effects were generated from a multivariate normal
### Table 7.8: Simulation results for 5 target distributions and correlation of covariates \( \rho = 0 \) (Mean Square Error \( \times 10^2 \)).

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(F)</th>
<th>(G)</th>
<th>(H)</th>
<th>(I)</th>
<th>(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lme4 ( \beta_{\theta_0} )</td>
<td>9.01 (0.31)</td>
<td>73.34 (7.03)</td>
<td>6.87 (0.78)</td>
<td>9.64 (0.46)</td>
<td>9.64 (1.23)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>8.13 (0.65)</td>
<td><strong>29.24</strong> (12.63)</td>
<td>6.94 (0.84)</td>
<td><strong>6.71</strong> (0.75)</td>
<td>9.32 (1.23)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.3324</td>
<td>0.0034</td>
<td>0.4631</td>
<td>0.0964</td>
<td>0.4591</td>
</tr>
<tr>
<td>lme4 ( \beta_{\theta_1} )</td>
<td>6.08 (0.42)</td>
<td>57.52 (8.75)</td>
<td>3.36 (0.48)</td>
<td>4.99 (0.39)</td>
<td>3.29 (0.31)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>6.25 (0.51)</td>
<td><strong>24.75</strong> (7.75)</td>
<td>3.28 (0.53)</td>
<td>4.99 (0.34)</td>
<td>3.63 (0.45)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.8542</td>
<td>0.0065</td>
<td>0.4833</td>
<td>0.9663</td>
<td>0.3200</td>
</tr>
<tr>
<td>lme4 ( \beta_{\theta_2} )</td>
<td>5.08 (0.23)</td>
<td>81.75 (9.10)</td>
<td>5.65 (1.11)</td>
<td>3.06 (0.28)</td>
<td>4.75 (0.34)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>5.16 (0.31)</td>
<td><strong>28.89</strong> (9.73)</td>
<td>5.73 (1.12)</td>
<td>3.37 (0.40)</td>
<td>4.61 (0.72)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.8545</td>
<td>0.0002</td>
<td>0.3254</td>
<td>0.6843</td>
<td>0.9120</td>
</tr>
<tr>
<td>lme4 ( \beta_{\theta_3} )</td>
<td>2.72 (0.12)</td>
<td>64.05 (8.33)</td>
<td>5.34 (0.68)</td>
<td>3.93 (0.29)</td>
<td>3.56 (0.23)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>2.34 (0.29)</td>
<td><strong>26.84</strong> (10.51)</td>
<td>5.21 (1.50)</td>
<td>3.88 (0.36)</td>
<td>3.23 (0.46)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.6115</td>
<td>0.0071</td>
<td>0.8765</td>
<td>0.7831</td>
<td>0.4823</td>
</tr>
<tr>
<td>lme4 ( \beta_{\theta_4} )</td>
<td>2.59 (0.23)</td>
<td>64.00 (8.66)</td>
<td>5.46 (0.48)</td>
<td>6.36 (0.42)</td>
<td>6.74 (0.54)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>2.52 (0.28)</td>
<td><strong>30.34</strong> (9.83)</td>
<td>5.87 (0.34)</td>
<td>7.01 (0.43)</td>
<td>6.32 (0.39)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.8104</td>
<td>0.0125</td>
<td>0.8459</td>
<td>0.3298</td>
<td>0.5871</td>
</tr>
<tr>
<td>lme4 ( \beta_1 )</td>
<td>3.50 (0.14)</td>
<td>6.25 (1.58)</td>
<td>3.96 (0.38)</td>
<td>2.35 (0.18)</td>
<td>2.78 (0.11)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>3.22 (0.28)</td>
<td><strong>1.81</strong> (0.31)</td>
<td>4.04 (0.42)</td>
<td>2.46 (0.23)</td>
<td>2.46 (0.28)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.2156</td>
<td>0.0088</td>
<td>0.7856</td>
<td>0.8430</td>
<td>0.2389</td>
</tr>
<tr>
<td>lme4 ( \beta_2 )</td>
<td>2.03 (0.11)</td>
<td>9.59 (2.21)</td>
<td>6.48 (1.07)</td>
<td>3.28 (0.24)</td>
<td>3.11 (0.10)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>2.14 (0.20)</td>
<td><strong>3.45</strong> (0.81)</td>
<td>6.44 (0.92)</td>
<td>3.23 (0.32)</td>
<td>3.00 (0.29)</td>
</tr>
<tr>
<td>( p )-value</td>
<td>0.6368</td>
<td>0.0145</td>
<td>0.3556</td>
<td>0.8718</td>
<td>0.7645</td>
</tr>
</tbody>
</table>

For distribution I, random effects were generated from a multivariate skew-normal distribution with shape parameter \( \alpha = (4,4,4,4)^T \). However, such skewness are not very obvious in high dimensional space, as shown in Fig. 7.2, which explains the performance of these two methods for distribution I.

### 7.3 Real Data Set

We investigated the Contraception data, which is obtained from R packages `mlmRev` (Bates et al., 2014b). We fitted models using the CNMM-MS and `lme4` methods,
## 7.3. REAL DATA SET

<table>
<thead>
<tr>
<th>Estimator</th>
<th>(F)</th>
<th>(G)</th>
<th>(H)</th>
<th>(I)</th>
<th>(J)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\rho = 0.9)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\beta_{\theta_0})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>13.32 (1.06)</td>
<td>89.70 (3.2)</td>
<td>12.05 (0.11)</td>
<td>12.24 (0.13)</td>
<td>14.04 (0.26)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>8.60 (0.63)</td>
<td>48.24 (3.7)</td>
<td>12.14 (0.13)</td>
<td>12.65 (0.16)</td>
<td>12.21 (0.32)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.0613</td>
<td>0.0001</td>
<td>0.8930</td>
<td>0.3591</td>
<td>0.0712</td>
</tr>
<tr>
<td>(\beta_{\theta_1})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>32.04 (2.71)</td>
<td>204.88 (12.40)</td>
<td>14.55 (0.63)</td>
<td>13.80 (0.54)</td>
<td>15.33 (0.34)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>25.53 (1.48)</td>
<td>45.84 (3.4)</td>
<td>13.39 (0.78)</td>
<td>10.40 (0.84)</td>
<td>14.42 (0.28)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.2959</td>
<td>7.585 \times 10^{-08}</td>
<td>0.4582</td>
<td>0.2190</td>
<td>0.3290</td>
</tr>
<tr>
<td>(\beta_{\theta_2})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>29.93 (2.47)</td>
<td>162.34 (9.25)</td>
<td>23.73 (1.21)</td>
<td>10.22 (0.79)</td>
<td>14.28 (0.42)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>31.30 (3.03)</td>
<td>154.31 (12.34)</td>
<td>21.45 (1.13)</td>
<td>8.86 (0.71)</td>
<td>15.12 (0.79)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.8623</td>
<td>0.7156</td>
<td>0.2721</td>
<td>0.3821</td>
<td>0.7311</td>
</tr>
<tr>
<td>(\beta_{\theta_3})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>35.05 (4.65)</td>
<td>130.10 (3.42)</td>
<td>13.80 (0.46)</td>
<td>14.91 (0.60)</td>
<td>16.58 (0.35)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>16.46 (1.47)</td>
<td>109.77 (9.27)</td>
<td>13.21 (0.57)</td>
<td>12.09 (0.78)</td>
<td>14.21 (0.43)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.0620</td>
<td>0.3561</td>
<td>0.8821</td>
<td>0.3581</td>
<td>0.0339</td>
</tr>
<tr>
<td>(\beta_{\theta_4})</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>19.81 (1.82)</td>
<td>122.18 (9.06)</td>
<td>15.20 (0.69)</td>
<td>11.34 (0.53)</td>
<td>14.31 (0.67)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>9.88 (0.71)</td>
<td>68.75 (6.32)</td>
<td>16.91 (0.81)</td>
<td>10.11 (0.69)</td>
<td>13.23 (0.48)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.0140</td>
<td>0.0001</td>
<td>0.6911</td>
<td>0.7842</td>
<td>0.2499</td>
</tr>
<tr>
<td>(\beta_1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>20.04 (3.55)</td>
<td>20.19 (1.32)</td>
<td>11.82 (0.54)</td>
<td>12.65 (0.32)</td>
<td>10.57 (0.43)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>9.84 (0.71)</td>
<td>20.11 (1.91)</td>
<td>13.41 (0.67)</td>
<td>13.12 (0.41)</td>
<td>8.81 (0.31)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.0560</td>
<td>0.9172</td>
<td>0.4391</td>
<td>0.7260</td>
<td>0.0810</td>
</tr>
<tr>
<td>(\beta_2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>27.42 (1.20)</td>
<td>22.35 (1.61)</td>
<td>18.08 (0.91)</td>
<td>12.21 (1.12)</td>
<td>9.00 (0.78)</td>
</tr>
<tr>
<td>CNMM-MS</td>
<td>15.24 (1.21)</td>
<td>20.60 (2.11)</td>
<td>17.32 (1.12)</td>
<td>12.30 (0.98)</td>
<td>8.58 (0.66)</td>
</tr>
<tr>
<td>(p-value)</td>
<td>0.0005</td>
<td>0.6590</td>
<td>0.8381</td>
<td>0.9315</td>
<td>0.5387</td>
</tr>
</tbody>
</table>

Table 7.9: Simulation results for 5 target distributions and correlation of covariates \(\rho = 0.9\) (Mean Square Error \(\times 10^2\)).

respectively, and compared the log-likelihood values of the fitted models and the estimates of fixed effects.

The Contraception data comes from the 1988 Bangladesh Fertility Survey about the use of contraception by women in urban and rural areas. It contains 1934 observations with 60 groups, and 6 variables: woman, the identifying code for each woman; district, the identifying code for each district; use, the contraceptive use at time of survey; livch, the number of living children at time of survey; age, the age of each woman; and residence, a binary variable indicating the type of region.
of residence.

Two random effects

<table>
<thead>
<tr>
<th>Estimator</th>
<th>$\theta_j^T$</th>
<th>$\pi_j$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$l(G_0, \theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMM-MS</td>
<td>$(-0.82, 0.92)$</td>
<td>0.30</td>
<td>$-0.01$</td>
<td>0.20</td>
<td>$-1524.16$</td>
</tr>
<tr>
<td></td>
<td>$(-0.48, 0.40)$</td>
<td>0.31</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$(-0.45, 0.49)$</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.28, 0.13)</td>
<td>0.34</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2.44, -1.16)</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean ($\theta$)</td>
<td>$(-0.25, 0.43)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.D. ($\theta$)</td>
<td>(0.61, 0.40)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>Mean ($\theta$)</td>
<td>(0.14, 0.06)</td>
<td>$-0.01$</td>
<td>0.20</td>
<td>$-1526.97$</td>
</tr>
<tr>
<td></td>
<td>S.D. ($\theta$)</td>
<td>(0.38, 0.25)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.10: Results of two estimators for Contraception data with two random effects.

Firstly, we consider fitting a GLMM model with two random effects. We treat use as the response variable, and both the intercept and residence as random effects. Here age and livch are fixed effects, where district provides the group information. The choices of fixed and random effects are basically arbitrary, since the principal aim here is to illustrate operation of the CNMM-MS. The mixed model studied here is

$$\logit(p_{ij}) = a_i + b_i {\text{residence}}_{ij} + \beta_1 {\text{age}}_{ij} + \beta_2 {\text{livch}}_{ij},$$

where $\text{residence}_{ij}$ denotes the random residence covariate of observation $i$ in district $j$, both $\text{age}_{ij}$ and $\text{livch}_{ij}$ are fixed effects covariates of observation $i$ in district $j$, while $a_i$ and $b_i$ are assumed to vary over different districts.

Table 7.10 provides the estimation results of the two methods. It can be seen that the CNMM-MS performs better than the lme4 in terms of the log-likelihood values. Also, the estimates of fixed effects are very similar for both methods.
### Three random effects

<table>
<thead>
<tr>
<th>Estimator</th>
<th>( \hat{\theta}_j^T )</th>
<th>( \hat{\pi}_j )</th>
<th>( \hat{\beta} )</th>
<th>( l(\hat{G}, \hat{\beta}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMM-MS</td>
<td>(-0.82, 0.95, -0.01)</td>
<td>0.25</td>
<td>0.20</td>
<td>-1524.16</td>
</tr>
<tr>
<td></td>
<td>(-0.51, 0.86, 0.02)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean (( \hat{\theta} ))</td>
<td>(-0.48, 0.41, -0.02)</td>
<td>0.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.30, 0.12, -0.01)</td>
<td>0.34</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2.44, -1.17, -0.01)</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S.D. (( \hat{\theta} ))</td>
<td>(-0.25, 0.42, -0.01)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lme4</td>
<td>Mean (( \theta ))</td>
<td>(-0.27, 0.43, -0.01)</td>
<td>0.20</td>
<td>-1526.45</td>
</tr>
<tr>
<td></td>
<td>S.D. (( \theta ))</td>
<td>(0.37, 0.25, 0.00)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.11: Results of two estimators for Contraception data with three random effects.

We also studied the data by assuming three random effects. We treat `use` as the response variable, and the intercept, `residence` and `age` as random effects. Therefore, only `livch` is a fixed effect in this case. Here `district` again provides group information. The mixed model here becomes

\[
\text{logit}(p_{ij}) = a_i + b_{1i} \text{residence}_{ij} + b_{2i} \text{age}_{ij} + \beta \text{livch}_{ij},
\]

where `residence_{ij}` and `age_{ij}` denote the random residence covariate and the random age covariate of observation \( i \) in district \( j \), and `livch_{ij}` the only fixed effect covariate of observation \( i \) in district \( j \), while \( a_i \), \( b_{1i} \) and \( b_{2i} \) are assumed to vary over different districts. Similarly, we arbitrarily chose the fixed and random effects here, to compare the performance of the CNMM-MS with the `lme4`.

Table 7.11 gives the results of the two methods. In the situation of three random effects, the CNMM-MS still performs better than the `lme4` in terms of log-likelihood values. The estimates of both the mean of random effects and fixed effects are very similar for both methods.

It can be seen that the log-likelihood value of the CNMM-MS in Table 7.11 is the same with that in Table 7.10. Moreover, although `age` in Table 7.11 is treated
as a random effect, the estimate coefficient of \textit{age} is almost a constant among all mixture components which has the similar value to that in Table 7.10 where \textit{age} is treated as a fixed effect. The estimates of intercept and \textit{residence} are almost the same in both Table 7.11 and Table 7.10, where they are treated as random effects.

\section*{7.4 Summary}

In this chapter, we study GLMM’s with multiple random effects by using semi-parametric mixtures. In particular, we make no assumption about the distribution of the random effects and use the non- and semiparametric maximum likelihood estimation method to estimate the distribution. An algorithm is developed based on the CNMMB and the BFGS algorithm. The proposed method is general to all GLMM’s, although we take binomial distributions with the logistic link function as an example in our investigation.

Without any assumption made for random effects, the new method is more flexible and able to always obtain more accurate estimates and it is thus advantageous over the method based on normality assumption. Particularly, when the distribution of random effects departs far from normality, numerical studies show that the new method outperforms the parametric method and this becomes more obvious when covariates are correlated.
Summary and Future Works

8.1 Summary

In this thesis, we provide a general method to fitting multivariate nonparametric and semiparametric mixtures, and apply them to density estimation. The new method performs significantly better than the kernel-based methods in both simulated and real data examples. Based on the idea, new methods are proposed to solve the problems of classification and generalized linear mixed-effects models.

In Chapter 2, previous work on nonparametric density estimation is reviewed. Particularly, kernel density estimation, especially the bandwidth selection methods, is introduced, followed by the review of mixture models including finite, nonparametric and semiparametric mixtures. The maximum likelihood methods for estimating mixture models are described, and the applications of mixture models to density estimation are also discussed. This chapter also introduces some usage of density estimation in statistical methods such as classification and generalized linear mixed models. Finally, some algorithms for fitting mixtures are briefly described.

In Chapter 3, a new method is proposed for nonparametric multivariate density estimation, which extends a general framework in the univariate case based on nonparametric and semiparametric mixture distributions. The major challenge to a
multivariate extension is the dilemma that one can not maximise directly the likelihood function with respect to the whole component covariance matrix, since the likelihood is unbounded for a singular covariance matrix, and that one can not leave the covariance matrix or a large part of it to be determined by a selection method, since it would be computationally infeasible. To overcome it, we use a volume parameter $h$ to enforce a minimal restriction on the covariance matrix so that, with $h$ fixed, the likelihood function is bounded and its maximization can be successfully carried out with respect to all the remaining parameters. The role played here by the scalar $h$ is just the same as by the bandwidth in the univariate case and its value can be determined by a model selection criterion, such as the Akaike information criterion. New efficient algorithms are also described for finding the maximum likelihood estimates of these mixtures under various restrictions on the covariance matrix.

In Chapter 4, empirical studies using simulated and real-world data show that the new multivariate mixture-based density estimator performs remarkably better than kernel-based density estimators.

In Chapter 5, a new density-based classification method that uses semiparametric mixtures is described. Like other density-based classifiers, it first estimates, with a semiparametric mixture, the probability density function for the observations in each class and then classifies a new observation by the highest posterior probability. By making a proper use of a multivariate nonparametric density estimator that has been developed in Chapter 3, it is able to produce adaptively smooth and complicated decision boundaries in a high-dimensional space and can thus work well in such cases. Issues specific to classification are studied and discussed. Numerical studies using simulated and real-world data show that the new classifier performs extremely well as compared with other commonly used classification methods.

Chapter 6 studies the generalized linear mixed models with multiple random effects. To fit a GLMM, a semiparametric multivariate mixture is applied to find
the NPMLE of the distribution of random effects and the MLE of the fixed effects. The proposed method effectively avoids the misspecification of the distribution of random effects, since it is based on nonparametric density estimation instead of the parametric assumption on the distribution. A new algorithm for computing a semiparametric MLE for a GLMM is also proposed. Moreover, the direct search directed derivative method for computing a semiparametric MEL is also studied. The difference between the proposed method and the direct search directed derivative method for fitting semiparametric mixtures are investigated.

In Chapter 7, simulation studies provide strong evidence that the new method performs significantly better than the parametric method in terms of the squared errors of the estimates of fixed effects. Real examples show that the new method outperforms the parametric method in estimating the random effects in terms of the log-likelihood values.

The thesis provides a new method to nonparametric distribution estimation in the multivariate situation. As is demonstrated in the thesis, the success in density estimation can be instrumented for solving other statistical problems, such as in the case of classification and mixed effects models that are studied in the thesis. Many statistical methods are based on the knowledge, typically gained from data, of some distribution functions. Therefore, with our new density estimation method available, new solutions can likely be found to these statistical problems. Future research will look into more of these problems.

8.2 Future Works

In this thesis, we provide a new procedure by using multivariate nonparametric and semiparametric mixtures to estimating a density function. It offers a new template for solving other statistical problems via nonparametric-mixture-based density estimation. Further investigations in this direction can be truly fruitful. Here are a few
Dealing with Discrete Data

The current work in Chapter 3 focuses on estimating a density function with continuous variables. For data mixed with continuous and discrete variables, it might be feasible to use truncated or wrapped normal distributions for mixture component distributions, with proper discretization for discrete variables. This hopefully can provide correlations among all variables in a convenient framework.

Cluster Analysis

Clustering, known as unsupervised learning, deals with the identification of latent groups of observations that are cohesive. Finite mixture models have already been used for clustering, which is called model-based clustering. The general logic of model-based clustering is to treat each of the subpopulation separately and model the overall population as a mixture of these subpopulations by using finite mixture models. It goes back at least to the work of Wolfe (1963), and has been reviewed by Fraley and Raftery (2002) and also well discussed by Hastie et al. (2009), chapter 14. Nonparametric density estimation has also been used for this task, when a cluster is often identified by a mode of the estimated density. The KDE is widely used for this purpose (Minnotte and Scott, 1993; Cheng et al., 2004; Li et al., 2007). However, the literature on mode-based clustering focuses primarily on univariate and bivariate data due to the computational complexity in the multivariate case. The MDE can straightforwardly be applied to cluster analysis. With this method, a density function can be estimated via a nonparametric or semiparametric mixture, and a cluster can be identified by either a mixture component or a mode of the density estimate. By sliding the value of the volume parameter, a hierarchy of clusters can be produced.
**Mode Hunting**

Mode hunting is about identifying modes of density function. It is useful in, e.g., the mode-based clustering where each mode represents a cluster. Previous work focused on the use of kernel density estimation, and hunting a mode were usually done by a numerical method, e.g., the Euler difference method (Leung et al., 2000) or the modal EM algorithm (Li et al., 2007). The MDE can be naturally used for mode hunting due to its mixture nature. By controlling the volume parameter, a hierarchical structure of modes can be constructed.
Bibliography


Dumbgen, L. and K. Rufibach (2009). Maximum likelihood estimation of a log-


multivariate kernel density estimation. *Scandinavian Journal of Statistics* 32,
485–506.


Fedorov, V. V. (1972). *Theory of Optimal Experiments (Engl. transl.).* New York:
Academic Press.

likelihood estimation and the generalized likelihood ratio when the true parameter
is on the boundary of the parameter space. *Statistics and Probability Letters* 13,
325–332.

*Annals of Eugenics* 7, 179–188.

Fix, E. and J. Hodges (1951). Discriminatory analysis, nonparametric discrimina-
tion: Consistency properties. Technical report, School of Aviation Medicine, U.S.
Air Force.


Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis,


Bibliography


