The Pennsylvania State University
The Graduate School
Department of Statistics

NONPARAMETRIC ESTIMATION IN
MULTIVARIATE FINITE MIXTURE MODELS

A Dissertation in
Statistics
by
Tatiana A. Benaglia

© 2008 Tatiana A. Benaglia

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2008
The dissertation of Tatiana A. Benaglia was reviewed and approved* by the following:

Thomas P. Hettmansperger
Professor of Statistics
Dissertation Adviser
Chair of Committee

Bruce G. Lindsay
Willaman Professor of Statistics
Head of the Department of Statistics

David R. Hunter
Associate Professor of Statistics

Hoben Thomas
Professor of Psychology

* Signatures are on file in the Graduate School.
Abstract

The main goal of this thesis is to provide a complete methodology to analyze finite multivariate mixture models. Our approach is fully nonparametric and it only requires the coordinates to be independent, conditional on the component membership. Since the number of components in the mixture is assumed to be known, we also developed a technique to select the number of components that best fits the model. In addition, we provide some tools to check if the assumption of conditional independence is reasonable. All the methods are evaluated by simulations, and compared with methodology existing in the literature. We also apply our methodology to two real datasets from cognitive psychology.
# Table of Contents

List of Tables ................................................................. vii

List of Figures ................................................................. xi

Acknowledgments ............................................................... xiv

Chapter 1. Introduction ...................................................... 1
  1.1 Finite Mixture Models .............................................. 1
  1.2 Identifiability and Previous Work ................................. 3
  1.3 EM Algorithm and the Missing Data Framework ................. 5
  1.4 Semiparametric Stochastic EM Algorithm of Bordes et al. (2007) 8

Chapter 2. EM-like Algorithm for Multivariate Mixtures ........... 12
  2.1 Introduction ......................................................... 12
  2.2 Nonparametric Multivariate Finite Mixture ....................... 13
  2.3 Nonparametric Estimation in Multivariate Mixtures .......... 15
      2.3.1 The Nonparametric EM (npEM) Algorithm ............... 16
  2.4 Bandwidth and Kernel Selection .................................. 18
  2.5 Modifications to the General Model and Algorithm .......... 22
      2.5.1 Location-Scale Model ...................................... 22
      2.5.2 Symmetric Components .................................... 24
      2.5.3 Changing Block Structure ................................ 25
  2.6 Selecting the Number of Components ............................. 26
      2.6.1 Kolmogorov-Smirnov Minimum Distance Method .......... 26
      2.6.2 Simulation Study ........................................... 30
      2.6.3 Choosing the Number of Components for the Water-level Task Data .................................................. 32
      2.6.4 Choosing the Number of Components for the RT Task Data 34
Chapter 3. Monte Carlo Simulations ........................................... 36
  3.1 Introduction ......................................................... 36
  3.2 Simulation Models and Results .................................... 36
      3.2.1 Normal Location Mixture with Conditionally iid Coordinates 38
      3.2.2 Laplace Location Mixture with Conditionally iid Coordinates 41
      3.2.3 Normal Scale Mixture with Conditionally iid Coordinates ... 43
      3.2.4 Normal Location Mixture with Coordinates not Identically
          Distributed ......................................................... 45
      3.2.5 Mixture of Asymmetric Distributions ............................ 49
      3.2.6 Mixture of Mixed Distributions ................................. 52
      3.2.7 Mixture with Blocks of Conditionally iid Coordinates ....... 54
      3.2.8 Comparing the npEM with the inversion method of Hall et al.
          (2005) .......................................................... 57
  3.3 Discussion ............................................................ 60

Chapter 4. Checking the Conditional Independence Assumption .......... 61
  4.1 Introduction ......................................................... 61
  4.2 Deriving the Correlation in Finite Mixture Models ................. 62
  4.3 Weighted Bootstrap Hypothesis Test ................................ 68
      4.3.1 Sampling from a Kernel Density Estimate .................... 69
      4.3.2 Smoothed Weighted Bootstrap ................................. 71
      4.3.3 Bootstrapped p-value ........................................ 72
      4.3.4 Goodness of fit tests ........................................ 72
  4.4 Simulation with Known Null Model ................................ 75
  4.5 Simulation Using the Weighted Smoothed Bootstrap ............... 78
  4.6 Comparing $\Sigma$ and $\Sigma_0$ using their eigenvalues .......... 80
  4.7 Checking the Conditional Independence Assumption in Real Data .. 81
      4.7.1 Waterdata ..................................................... 81
      4.7.2 RTdata ....................................................... 83
Chapter 5. Data Analysis .................................................. 85
  5.1 Introduction ....................................................... 85
  5.2 Waterdata .......................................................... 85
  5.3 RTdata .............................................................. 90

Chapter 6. Discussion and Future Work ................................. 95
  6.1 Introduction ....................................................... 95
  6.2 Discussion ........................................................ 95
  6.3 Future Work ....................................................... 97

Appendix A. Extra Simulations Results from Chapter 4 .............. 99
  A.1 Multivariate Normal Location Mixture in 3.2.4 ................. 99
  A.2 Mixture of Asymmetric Distributions in 3.2.5 .................. 101
  A.3 Mixture of Mixed Distributions in 3.2.6 ....................... 103

Appendix B. mixtools: Short Tutorial for Nonparametric Mixtures . 105
  B.1 Generating and analyzing a multivariate mixture in mixtools . 105

References ........................................................................... 110
# List of Tables

1.1 MSE for the estimates for spSEM and spEM. ........................................ 11  
1.2 Bias of the estimates for spSEM and spEM. ........................................ 11  
2.1 Notation. ............................................................................................... 15  
2.2 Values of factor in $c_n$ as a function of the number of components $m$. ... 28  
2.3 Minimum number of repeated measures for identifiability. ...................... 29  
2.4 Accuracy of the Kolmogorov-Smirnov Minimum Distance Method. ... ... 31  
2.5 Selecting the number of components for a two component normal location  mixture (Model 1 with $m = 2$). ......................................................... 31  
2.6 Selecting the number of components for a three component normal scale  mixture (Model 2 with $m = 3$). ............................................................ 32  
2.7 Selecting the number of components for a four component normal location  mixture (Model 3 with $m = 4$). ............................................................ 32  
2.8 Selecting the number of components for the Waterdata when assuming  non identically distributed coordinates....................................................... 33  
2.9 Selecting the number of components for the Waterdata with all 8 coordinates grouped into 4 blocks. ............................................................... 34  
2.10 Selecting the number of components for the Waterdata when assuming  conditionally iid coordinates............................................................... 34  
2.11 Selecting the number of components for the RTdata when assuming non  identically distributed coordinates....................................................... 35  
2.12 Selecting the number of components for the RTdata when assuming condi-  tionally iid coordinates............................................................... 35  
3.1 Two-component normal mixture with $\lambda = 0.5, \mu_1 = 0, \mu_2 = 2$, and  $\sigma_1^2 = \sigma_2^2 = 1$. The results are based on 1000 simulations with sample size  $n = 100$ each. ............................................................... 39
3.2 Two-component normal mixture with $\lambda = 0.25, \mu_1 = 0, \mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 1$. The results are based on 1000 simulations with sample size $n = 100$ each. ........................................ 40

3.3 Two-component Laplace mixture with $\lambda = 0.25, \mu_1 = 0, \mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 2$. The results are based on 1000 simulations with sample size $n = 100$ each. ........................................ 42

3.4 Two-component normal mixture with $\lambda = 0.5, \mu_1 = \mu_2 = 0, \sigma_1^2 = 1$, and $\sigma_2^2 = 9$. The results are based on 1000 simulations with sample size $n = 100$ each. ........................................ 44

3.5 Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.3$. The results are based on 500 simulations with sample size $n = 500$ each. .... 46

3.6 MISE for the estimated density functions for each coordinate within each component in two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.3, 0.5$, and 0.8. ........................................ 46

3.7 Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (2, 2.5, 3)^t$, $\sigma_1^2 = (1, 1, 1)^t$, and $\sigma_2^2 = (1.5, 2, 1)$ when $\lambda = 0.3$. The results are based on 500 simulations with sample size $n = 500$ each. .... 48

3.8 MISE for the estimated density functions for each coordinate within each component in two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (2, 2.5, 3)^t$, $\sigma_1^2 = (1, 1, 1)^t$, and $\sigma_2^2 = (1.5, 2, 1)$ when $\lambda = 0.3, 0.5$, and 0.8. ........................................ 48

3.9 Two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2)$, and $\beta_2 = (2, 1, 0.5)$ for $\lambda = 0.5$. The means and variances are $\alpha\beta$ and $\alpha\beta^2$, respectively. The results are based on 1000 simulations with sample size $n = 300$ each. ........................................ 50

3.10 MISE for the estimated density functions for each coordinate within each component in a two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2)$, and $\beta_2 = (2, 1, 0.5)$ when $\lambda = 0.3, 0.5$, and 0.8. 51
3.11 Two-component mixture with the coordinates within a component having different distributions for $\lambda = 0.5$. The results are based on 1000 simulations with sample size $n = 300$ each. ............................................................... 53

3.12 MISE for the estimated densities for two component mixture with the coordinates within a component having different distributions for $\lambda = 0.3, 0.5, \text{and } 0.8$. ................................. 53

3.13 Three-component mixture with two blocks of conditionally iid coordinates and $\lambda = (0.25, 0.35, 0.4)$. The results are based on 1000 simulations with sample size $n = 300$ each. ................................. 55

3.14 MISE for the estimated density functions for a three component mixture with two blocks of conditionally iid coordinates and $\lambda = (0.25, 0.35, 0.4)$. ................................. 56

4.1 Estimated size and power of the tests from simulation. .................. 78

4.2 Estimated size and power of the tests from smoothed weighted bootstrap. 78

5.1 Waterdata: estimates and standard errors for the component means and variances for a three component mixture with four blocks of conditionally iid coordinates .............................. 91

5.2 RTdata: estimates and standard errors for the component means and std. deviations for a three component mixture with conditionally independent coordinates. .......................... 93

5.3 RTdata: estimates and standard errors for the component means and standard deviations for a three component mixture with conditionally iid coordinates. .......................... 93

A.1 Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t$, $\mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.5$. The results are based on 500 simulations with sample size $n = 500$ each. .......................... 99

A.2 Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t$, $\mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.8$. The results are based on 500 simulations with sample size $n = 500$ each. .......................... 100
A.3 Two-component multivariate normal mixture with \( \boldsymbol{\mu}_1 = (0, 0, 0)^t, \boldsymbol{\mu}_2 = (2, 2.5, 3)^t, \boldsymbol{\sigma}_1^2 = (1, 1, 1)^t, \) and \( \boldsymbol{\sigma}_2^2 = (1.5, 2, 1) \) when \( \lambda = 0.5 \). The results are based on 500 simulations with sample size \( n = 500 \) each. 

A.4 Two-component multivariate normal mixture with \( \boldsymbol{\mu}_1 = (0, 0, 0)^t, \boldsymbol{\mu}_2 = (2, 2.5, 3)^t, \boldsymbol{\sigma}_1^2 = (1, 1, 1)^t, \) and \( \boldsymbol{\sigma}_2^2 = (1.5, 2, 1) \) when \( \lambda = 0.8 \). The results are based on 500 simulations with sample size \( n = 500 \) each. 

A.5 Two-component gamma mixture with \( \boldsymbol{\alpha}_1 = (2, 2, 2), \boldsymbol{\alpha}_2 = (5, 10, 15), \boldsymbol{\beta}_1 = (2, 2, 2), \) and \( \boldsymbol{\beta}_2 = (2, 1, 0.5) \) for \( \lambda = 0.3 \). The means and variances are \( \alpha \beta \) and \( \alpha \beta^2 \), respectively. The results are based on 1000 simulations with sample size \( n = 300 \) each. 

A.6 Two-component gamma mixture with \( \boldsymbol{\alpha}_1 = (2, 2, 2), \boldsymbol{\alpha}_2 = (5, 10, 15), \boldsymbol{\beta}_1 = (2, 2, 2), \) and \( \boldsymbol{\beta}_2 = (2, 1, 0.5) \) for \( \lambda = 0.8 \). The means and variances are \( \alpha \beta \) and \( \alpha \beta^2 \), respectively. The results are based on 1000 simulations with sample size \( n = 300 \) each. 

A.7 Two-component mixture with the coordinates within a component having different distributions for \( \lambda = 0.3 \). The results are based on 1000 simulations with sample size \( n = 300 \) each. 

A.8 Two-component mixture with the coordinates within a component having different distributions for \( \lambda = 0.8 \). The results are based on 1000 simulations with sample size \( n = 300 \) each.
List of Figures

2.1 Effect of bandwidth in the component density estimates. On the left we have estimates with the same bandwidth, and on the right with different bandwidths. ................................................................. 21

2.2 Clock orientation and tilted vessel illustrating the Water-level task. . . . . 33

3.1 Estimated component density and cdf for a two-component normal mixture with $\lambda = 0.5, \mu_1 = 0, \mu_2 = 2$, and $\sigma_1^2 = \sigma_2^2 = 1$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions. ................................. 40

3.2 Estimated component density and cdf for a two-component normal mixture with $\lambda = 0.25, \mu_1 = 0, \mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 1$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions. ................................. 41

3.3 Estimated component density and cdf for a two-component Laplace mixture with $\lambda = 0.5, \mu_1 = 0, \mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 2$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions. ................................. 43

3.4 Estimated component density and cdf for a two-component normal mixture with $\lambda = 0.5, \mu_1 = \mu_2 = 0, \sigma_1^2 = 1$, and $\sigma_2^2 = 9$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions. ................................. 44

3.5 Estimated component densities and cdfs for a two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.3$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions. ................................. 47
3.6 Estimated component densities and cdfs for a two-component multivariate normal mixture with \( \mu_1 = (0, 0, 0)^t, \mu_2 = (2, 2.5, 3)^t \), and \( \sigma_1^2 = (1, 1, 1)^t \), and \( \sigma_2^2 = (1.5, 2, 1)^t \) when \( \lambda = 0.3 \). The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

3.7 Estimated component densities and cdfs for a two-component gamma mixture with \( \alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2), \) and \( \beta_2 = (2, 1, 0.5) \) when \( \lambda = 0.5 \). The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

3.8 Estimated component densities and cdfs for a two-component mixture with the coordinates within a component having different distributions when \( \lambda = 0.5 \). The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

3.9 Component densities and cdfs for a three-component mixture with blocks

3.10 Square roots of MISEs on a logarithmic scale as a function of \( \lambda \), for each model for all \( f_{jk}, j = 1, 2 \) and \( k = 1, 2, 3 \).

4.1 Bivariate Normal Mixtures - Density Functions

4.2 Bivariate Normal Mixtures - Density Contours

4.3 Histogram of the p-values under the null model when using simulation. Each row of histograms represents a different test, and each column is for a number of repeated measure \( (r = 3, 5, \) and 8). Histograms for \( M, KS, \) and \( CvM \) tests are in first, second, and third row, respectively.

4.4 Histogram of the p-values under the null model when using the smoothed weighted bootstrap. Each row of histograms represents a different test, and each column is for a number of repeated measure \( (r = 3, 5, \) and 8). Histograms for \( M, KS, \) and \( CvM \) tests are in first, second, and third row, respectively.
4.5 Bootstrap 95% CIs for the average ratio of the eigenvalues for $S$ and $S_0$. On the left we have the CIs when the data has conditionally independent coordinates, and on the right we have the conditionally non-independent case. The dashed line represents our target value 1. Each dataset have $n = 300$ observations and $r = 3$ repeated measures.

4.6 Eigenvalues of $S$ and $S_0$ computed for the Waterdata.

4.7 Sampling distribution of $M$ and $KS$ test statistics. The dashed line represents the observed value of $M$ and $KS$ computed from the Waterdata.

4.8 Eigenvalues of $S$ and $S_0$ computed for the RTdata.

4.9 Sampling distribution of $M$ and $KS$ test statistics. The dashed line represents the observed value of $M$ and $KS$ computed from the RTdata.

5.1 Waterdata component pdfs and cdfs for a four component mixture assuming conditionally iid coordinates.

5.2 Waterdata component cdfs for a three component mixture with conditionally independent coordinates.

5.3 Waterdata component cdfs for a three component mixture with four conditionally independent blocks of coordinates.

5.4 Waterdata component pdfs for a three component mixture with four conditionally independent blocks of coordinates.

5.5 RTdata component cdfs for a three component mixture with conditionally independent coordinates.

5.6 RTdata component pdfs and cdfs for a three component mixture with conditionally iid coordinates.

B.1 Component pdfs for a two component mixture of normals with conditionally independent coordinates.

B.2 Component pdfs for a two component mixture of normals with blocks of conditionally iid coordinates.
Acknowledgments

To Tom Hettmansperger, my thesis advisor, who I cannot thank enough for all of his advice and guidance. Your help and expertise made me enjoy even more working in this field. I will truly miss our meetings, where besides talking about our research, you always had a story to tell. You played a very important role in my life through these four years of graduate school. You are an amazing teacher, a friend always ready to listen, a father figure, and an example to follow.

I would like to thank the other members of my committee: Bruce Lindsay, for sharing his expertise in mixture models; Hoben Thomas, for providing the dataset that motivated a great part of this research; and David Hunter, jointly with Didier Chauveau, for their direct and fundamental contributions to this thesis. It has been a pleasure to work with you both.

Most importantly, I want to thank my entire family, especially my parents, Adauto and Alzira, my sister Taís, and my grandma Albertina (in memoriam). We know how much we missed each other, and how hard it was to be so far away, but we were always close in our hearts. Thanks for always supporting my decisions. Your love made me stronger to reach this very important goal in my life. I love you all.

An enormous thanks to Benilton Carvalho, a very special person in my life. It is easy to accept new challenges and discover new worlds when I have someone like you by my side. You are my friend, my partner, my R guru. Thanks for your unconditional love and support. Also thanks to his parents, Nasaré and Benedito, for their contribution when I was applying for the scholarship that allowed me to go to graduate school.

To my American family, friends that I met during graduate school, who made my experience abroad unforgettable: Tracey Wrobel, Ruth Hummel, Megan and Chuck Romer, Risé Lara, and Ben Hammel. Thanks Tracey for being this friend who I can count on at any time and, jointly with Ben, did so much for me, especially while I was finishing this thesis. Ruth, thanks for being my roommate and personal English tutor,
always making fun of me at first, but teaching me a lot. Thanks Risé, my roommate during these four years, and Megan, a very special friend. I will miss our weekly dinners.

To the Brazilian community I met here at Penn State and at Johns Hopkins, thanks all of you for helping me feel less home-sick. I would like to also thank Samara Kiihl, someone that I have known since college in Brazil, who was always good company and ready for a good movie every time I was in Baltimore. Thanks to Lourdes and Eduardo Santiago, my dear Mexicans, for your friendship and funny stories. And a very special thanks to Anna Mazzucato for being so generous. I will never forget what you have done for me.

I would like to express my gratitude to my sponsor, the Brazilian Ministry of Education - CAPES, for funding these four years of graduate school, the Department of Statistics at Penn State for the teaching assistantship, and the National Science Foundation (NSF) grant for the research assistantship.

Thanks to all the professors in the Department of Statistics, who taught me so many important concepts and enriched my statistical background. Also, thanks to all of our staff for always being so helpful whenever I needed.

Last, but not least, thank God for allowing me to accomplish one more goal in life.
To my parents, Adauto and Alzira, and to Benilton.
Chapter 1

Introduction

1.1 Finite Mixture Models

Suppose the vectors $X_1, \ldots, X_n$ are an independent and identically distributed random sample from an $m$-component mixture distribution. The probability density function (pdf) of each $r$ dimensional $X_i$ is written as a convex combination of the $m$ individual density functions:

$$g_\varphi(x_i) = \sum_{j=1}^{m} \lambda_j \phi_j(x_i), \quad \text{with} \quad \sum_{j=1}^{m} \lambda_j = 1, \quad x_i \in \mathbb{R}^r, \quad (1.1)$$

where $\lambda_j \geq 0$ are the mixing proportions, and $\varphi^t = (\lambda^t, \phi^t) = (\lambda_1, \ldots, \lambda_m, \phi_1, \ldots, \phi_m)$ denotes the parameter vector to be estimated. When the $\phi_j$s are assumed to belong to a known parametric family $\mathcal{F} = \{f(\cdot | \xi), \xi \in \mathbb{R}^d\}$ indexed by an Euclidean $d$ dimensional parameter $\xi$, and the number of components $m$ is considered fixed, there are standard mixture techniques described in Titterington et al., 1985; Lindsay, 1995; McLachlan and Peel, 2000 to estimate the parameter vector $\varphi$ in model (1.1).

Choosing a parametric family $\mathcal{F}$ may be difficult. Sometimes, we do not have any information about the mixture component distributions and this is one of the motivations to adopt a nonparametric approach. We do not assume that the $\phi_j$s come from a family of densities that may be indexed by a finite-dimensional Euclidean parameter vector. Avoiding the parametric assumption is our goal in this thesis and this makes the model (1.1) more flexible than the parametric approach.

It is important to mention that model (1.1) is not identifiable if no restrictions are placed on $\phi_1, \ldots, \phi_j$, where by “identifiable” we mean that $g_\varphi$ has a unique representation of the form (1.1). In this thesis, we do not consider that a permutation of the $m$ pairs $(\lambda_1, \phi_1), \ldots, (\lambda_m, \phi_m)$, known as label-switching, produces a distinct representation.
In order to make model (1.1) identifiable, a common restriction placed on \( \phi_j(x) \), \( j = 1, \ldots, m \), which we adopt throughout this thesis, is that the coordinates of the \( \mathbf{X}_i \) vector are what we will call \textit{conditionally independent}, and by this we mean that the coordinates of the \( \mathbf{X}_i \) vector are independent, conditional on the subpopulation or component (\( \phi_1 \) through \( \phi_m \)) from which \( \mathbf{X}_i \) is drawn. Under this assumption, each joint density \( \phi_j(\cdot) \) is equal to the product of its marginal densities. Therefore, model (1.1) becomes

\[
g_\phi(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}),
\]

where \( f_{jk}(\cdot) \) denotes an unspecified univariate density function, which will be estimated either semi- or non-parametrically in this thesis. This model allows, for a given component, each coordinate of the \( \mathbf{X}_i \) to have a different density function. This is the most general model, but notice that when \( f_{jk} \) does not depend on \( k \), we have the case in which the coordinates of \( \mathbf{X}_i \) are not only conditionally independent but conditionally identically distributed as well. In this case, the model becomes:

\[
g_\phi(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{j}(x_{ik}),
\]

Further, in Section 2.2, we will describe the model introduced in Benaglia et al. (2008), which assumes that some of the coordinates can be grouped into \textit{blocks}, such that within a block, the coordinates are still conditionally independent but also identically distributed, and coordinates belonging to different blocks will have different density functions.

The motivation for our approach comes from the stochastic semiparametric EM algorithm introduced by Bordes et al. (2007). They extended the usual EM algorithm (Dempster et al., 1977) to estimate a semiparametric mixture in the univariate case. The semiparametric approach is due to the fact that they need to assume that \( f(\cdot) \) is a location-shifted symmetric distribution to make the model identifiable. We will give more details about this algorithm in Section 1.4. An extension of this algorithm to the
multivariate case is presented in this thesis. We will talk about identifiability results for multivariate finite mixtures in Section 1.2. For a given number of components and a minimum number of repeated measures that make the model identifiable, our approach is fully nonparametric, since we do not need any restrictions on the mixture component density functions $f_{jk}$.

1.2 Identifiability and Previous Work

Consider the model (1.2). This model is not identifiable unless some restrictions are imposed on the mixture component densities $f_{jk}$. One can ask how restrictive these assumptions would have to be in order to make this model identifiable. There is no easy answer to this question, but in this section we will discuss some previous work that is related to our problem and some relevant identifiability results.

In the univariate case ($r = 1$), Bordes et al. (2006) and Hunter et al. (2007) found that when $f_j(x) = f(x - \mu_j)$ for some density $f(\cdot)$ that is symmetric about zero, the mixture density $g_\phi(x)$ admits a unique representation whenever $m \leq 3$, except in certain special cases that are easily enumerable. For example, if $m = 2$ the location-shifted symmetric mixture is identifiable as long as $\lambda_1 \neq 1/2$.

In the multivariate case, identifiability for the general model (1.2) has not yet been settled, but it has been discussed by some authors. For example, in the two component case ($m = 2$), Hall and Zhou (2003) prove that model (1.2) is nonparametrically identifiable as long as each subject has at least three independent measurements ($r \geq 3$) under an additional assumption that they term “irreducibility” — namely, that none of the bivariate marginals of the mixture density factors into the product of its univariate marginals. Hall et al. (2005) extend these results and give a lower bound on $r$, as a function of $m$, that is necessary in order to guarantee identifiability. They state that $r$ and $m$ should satisfy $2^r - 1 \geq mr + 1$. As they say, this is the case where “from at least one point of view, the ‘curse of dimensionality’ works in reverse.” Although, Hall et al. (2005) do not give an explicit bound that is sufficient to guarantee identifiability, Elmore et al. (2005) prove that such a (finite) lower bound exists.
It is important to emphasize that we use the term “nonparametric” to describe the case in which no assumptions are made about the form of the \( f_{jk}(\cdot) \) even though the parameter vector \( \lambda \) is Euclidean. We reserve the term “semiparametric” for the case in which \( f_{jk}(\cdot) \) is partly specified by a finite-dimensional parameter, such as the case discussed above in which \( f_j(x) = f(x - \mu_j) \) for a symmetric but otherwise completely unspecified density \( f(\cdot) \). Note that Lindsay (1995) speaks of “nonparametric mixture modeling” in a different sense: The family \( \mathcal{F} \) from which the component densities come is fully specified up to a parameter vector \( \theta \), but the mixing distribution from which the \( \theta \) are drawn is assumed to be completely unspecified \textit{a priori}.

Several authors have recently addressed the problem of estimating the \( f_{jk} \) in model (1.2), but the estimation methods they propose appear to apply in only very limited cases. Hall et al. (2005), for example, give estimators based on inversion of mixture models that apply only to the case when \( m = 2 \) and \( r = 3 \), but analytical difficulties appear when applying this method beyond these cases. Even in the case \( r = 1 \), where restrictions as described at the beginning of this section must be placed on \( f_{j}(\cdot) \) in order to ensure identifiability, the estimation methods of Bordes et al. (2006) and Hunter et al. (2007) are difficult if not impossible to apply beyond the case \( m = 2 \). We discuss this case in Section 2.5.2.

By contrast, several authors have considered model (1.3) for continuous data that is, the case of conditionally independent and identically distributed coordinates. Hettmansperger and Thomas (2000); Cruz-Medina et al. (2004); Elmore et al. (2004) have developed the cutpoint approach to estimate this model. The cutpoint model discretizes the continuous measurements by replacing each observation \((x_{i1}, \ldots, x_{ir})\) by a multinomial vector \((n_1, \ldots, n_p)\), where

\[
n_a = \sum_{k=1}^{r} I\{c_{a-1} < x_{ik} \leq c_a\}, \quad 1 \leq a \leq p,
\]

and the cutpoints \(-\infty = c_0 < c_1 < \cdots < c_p = \infty\) are specified by the experimenter. The cutpoint approach is completely general in the sense that it can be applied to any number of components \( m \) and any number of repeated measures \( r \), just as long as \( r \geq 2m - 1 \), a
condition that guarantees identifiability (see Elmore and Wang, 2003; Kim, 1984). Some information is lost in the discretization step. Furthermore, even if the assumption of conditional independence is warranted, the extra assumption of identically distributed coordinates may not be; and the cutpoint method collapses when the coordinates are not identically distributed.

We take a different approach, which is also the method presented in Benaglia et al. (2008). We extend the univariate stochastic semiparametric algorithm of Bordes et al. (2007) to the multivariate conditionally independent case and eliminate the stochasticity. Our algorithm combines the best features of all the algorithms discussed previously: It is simple to program, it is applicable to any \( m \) and \( r \). We can estimate the most general model (1.2) as well as the one with conditionally independent and identically distributed coordinates represented by model (1.3)). In Section 2.2, we will see that we can even assume that only certain sets of coordinates are i.i.d., which we call blocks. We develop kernel-density-like estimates for each of the \( f_{jk} \), from which is possible to compute moments or any characteristic of interest.

Since our estimation method can be easily applied to an arbitrary number of coordinates or mixture components, unlike any previously published algorithms for this problem, we are in a position in which practice is more advanced than theory. Thus, it is prudent to exercise caution when trying to fit a model for which the identifiability question is not settled.

### 1.3 EM Algorithm and the Missing Data Framework

In this section we will discuss how the expectation-maximization (EM) algorithm of Dempster et al. (1977) can be used in the estimation of mixture problems. As mentioned in McLachlan and Peel (2000), the fitting of finite parametric mixture models by maximum likelihood had been studied in the past, but it was Dempster et al. (1977) who greatly stimulated interest in the use of finite mixture distributions to model heterogeneous data. This is because the EM algorithm simplifies the fitting of parametric mixture models by maximum likelihood, since the data can be viewed as being incomplete.
In our case, since we do not assume a parametric form for the component densities, we do not have a likelihood to maximize, but we will see that the missing data framework remains the same and the usual EM algorithm can still be applied. The difference is in the maximization step, which will be basically used to estimate the mixing proportions and, Bordes et al. (2007) propose the addition of one step where a density estimate is obtained (see details in Section 1.4).

Suppose the $r$ dimensional observation vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are realizations from iid random vectors with common mixture density given by (1.1), but we will assume that $\phi_j(\cdot) = f(\cdot|\theta_j)$, and that $f(\cdot|\theta_j)$ is from a specified parametric family. Hence, the parametric mixture is written as

$$ g_\varphi(x_i) = \sum_{j=1}^{m} \lambda_j f(x_i|\theta_j), \quad (1.4) $$

where $x_i \in \mathbb{R}^r$, and $\varphi = (\lambda_1, \ldots, \lambda_{m-1}, \theta_1^t, \ldots, \theta_m^t)^t$ is the vector containing all the unknown parameters. If we estimate the parameters by maximum likelihood, the observed-data likelihood function is

$$ L(\varphi) = \prod_{i=1}^{n} \sum_{j=1}^{m} \lambda_j f(x_i|\theta_j), \quad (1.5) $$

which leads to the log-likelihood function

$$ \ell(\varphi) = \log [L(\varphi|x)] = \sum_{i=1}^{n} \log \left[ \sum_{j=1}^{m} \lambda_j f(x_i|\theta_j) \right]. \quad (1.6) $$

It is not easy to find the MLE of $\varphi$ by maximizing (1.6) directly, but if the mixture is formulated as an incomplete-data problem, the EM algorithm by Dempster et al. (1977) can be used to estimate $\varphi$, as we will see next. The EM algorithm is essentially an iterative procedure iterating between the E-step (expectation) and the M-step (maximization) until convergence.
In mixture problems, we usually do not know from which subpopulation the observation \( X_i \) came. Then we define the missing data to be multinomial observations, \( Z_1, \ldots, Z_n \), which are indicators of component membership, i.e. the \( m \) dimensional vectors \( Z_1, \ldots, Z_n \) are distributed according to \( Mult(1, \lambda^t) \), where \( \lambda = (\lambda_1, \ldots, \lambda_m)^t \) is the vector of mixing proportions. For example, if we have a five component mixture and the first observation, \( X_1 \), belongs to the second component, then \( Z_1 = (0,1,0,0,0)^t \).

The complete-data is a \( n \times (r + m) \) matrix denoted by \( X_c = (X, Z) \), where \( X = (X_1 \ldots X_n)^t \), and \( Z = (Z_1 \ldots Z_n)^t \). Notice that if \( X_c \) were available, the estimation of the mixture distribution would be more straightforward, since each component density could be estimated directly from the data belonging to it. But \( Z_1, \ldots, Z_n \) are treated as latent vectors. In this case, the complete-data distribution is given by

\[
g_c^\varphi(x_i, z_i) = \prod_{j=1}^{m} \left[ \lambda_j f(x_i | \theta_j) \right]^{z_{ij}}, \tag{1.7}\]

and complete-data log-likelihood is

\[
\ell_c(\varphi) = \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij} \left[ \log(\lambda_j) + \log(f(x_i | \theta_j)) \right]. \tag{1.8}\]

In the E-step, we take the conditional expectation of \( \ell_c(\varphi|x) \) given the observed data matrix \( x \), using the current estimate for \( \varphi \), say \( \varphi^t \). Then on the \((t+1)\)th iteration of the EM algorithm, the E-step requires the calculation of:

\[
Q(\varphi; \varphi^t) \equiv E_{\varphi^t}[\ell_c(\varphi)|x]. \tag{1.9}\]

Since \( \ell_c(\varphi) \) is linear in \( z_{ij} \), it is only necessary to compute the conditional expectation of \( Z_{ij} \) given the observed data vector \( x_i \). Notice that the \( j \)th mixing proportion can be viewed as the prior probability that an observation belongs to the \( i \)th mixture component, that is, \( \lambda_j^t = P_{\varphi^t}(Z_{ij} = 1) \), for \( j = 1, \ldots, m \). Hence, using Bayes’ theorem,
we get
\[ p_{ij}^{t+1} = E_{\varphi^t}(Z_{ij} | x) = P_{\varphi^t}(Z_{ij} = 1 | x_i) = \frac{\lambda_{ij}^t f(x_i | \theta_j^t)}{\sum_{j' = 1}^{m} \lambda_{ij}^t f(x_i | \theta_{j'}^t)}, \] (1.10)

for \( j = 1, \ldots, m \), and \( i = 1, \ldots, n \).

Therefore, \( p_{ij}^{t+1} \) is also interpreted as the posterior probability that the \( i \)th observation vector belongs to the \( j \)th component given the observed data and the current parameter vector. Now using (1.10) we can take the conditional expectation of (1.8) given \( x \), which yields

\[ Q(\varphi; \varphi^t) = \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{t+1} \left[ \log(\lambda_j) + \log(f(x_i | \theta_j)) \right]. \] (1.11)

In the M-step, on the \((t+1)\)th iteration, it is required to maximize \( Q(\varphi; \varphi^t) \) with respect to \( \varphi \) to obtain the updated estimate \( \varphi^{t+1} \), and this solution also maximizes the observed data likelihood function in (1.5) (see Dempster et al., 1977). For the mixing proportions, since the \( z_{ij} \) are unobservable, they are replaced by \( p_{ij}^{t+1} \), and then an estimate of \( \lambda_j \) is computed by

\[ \lambda_{ij}^{t+1} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{t+1}, \quad \text{for } j = 1, \ldots, m. \] (1.12)

Given a set of initial or starting values \( \varphi^0 \), the E- and M-steps are alternated repeatedly until convergence.

The concepts and notation discussed in this section will be used throughout this thesis. In the semi- and non-parametric case we will see that is necessary to estimate the component densities and include this as an additional step in the iterative procedure.

1.4 Semiparametric Stochastic EM Algorithm of Bordes et al. (2007)

Here we will recall the semiparametric stochastic EM (spSEM) algorithm introduced by Bordes et al. (2007). They extended the usual EM algorithm in the missing
data setup (Dempster et al., 1977) to the semiparametric model by introducing one step of nonparametric density estimation of the unknown mixed pdf. In this paper, they describe the algorithm for the univariate location-shifted mixture model, since identifiability had been proved for the two and three components cases.

Suppose the observations $X_1, \ldots, X_n$ are a simple random sample from a $m$-component mixture distribution. The pdf of each $X_i$ is written as

$$g_\varphi(x_i) = \sum_{j=1}^{m} \lambda_j f_j(x_i), \quad \text{with} \quad \sum_{j=1}^{m} \lambda_j = 1, \quad \lambda_j \geq 0. \quad (1.13)$$

Bordes et al. (2006) and Hunter et al. (2007) proved that the model (1.13) is almost always identifiable when $f_j(x) = f(x - \mu_j)$ for some density $f(\cdot)$ that is symmetric about zero, whenever $m \leq 3$, except in certain special cases that are easily enumerable, as we discussed in Section 1.2. Thus, we have the symmetric location mixture model:

$$g_\varphi(x_i) = \sum_{j=1}^{m} \lambda_j f(x_i - \mu_j), \quad \text{with} \quad \sum_{j=1}^{m} \lambda_j = 1, \quad x_i \in \mathbb{R}, \quad (1.14)$$

where $\varphi = (\theta, f)$ denotes the parameter, which consists of both a parametric part $\theta^t = (\lambda^t, \mu^t)$ and a nonparametric part $f$, characterizing a semiparametric approach.

Given initial values $\varphi^0 = (\theta^0, f^0)$, we iterate the following steps:

1. **E-Step:** Calculate the posterior probabilities

   $$p_{ij}^t \overset{\text{def}}{=} E_{\varphi^t} (Z_{ij} | x_i) = P_{\varphi^t}(Z_{ij} = 1 | x_i) = \frac{\lambda_j^t f^t(x_i - \mu_j^t)}{\sum_{j'=1}^{m} \lambda_{j'}^t f^t(x_i - \mu_{j'}^t)}, \quad (1.15)$$

   for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

2. **Stochastic Step:** Simulate multinomial vectors such that:

   $$z_{i}^* = (z_{i1}^*, \ldots, z_{im}^*) \sim \text{Mult}(1, (p_{i1}^t, \ldots, p_{im}^t)). \quad (1.16)$$
In this step, the data is clustered in \( m \) groups, where the \( i \)th observation is assigned to the \( j \)th component with probability \( p_{ij}^{t} \).

3. **M-Step:** For \( j = 1, \ldots, m \), set

\[
\begin{align*}
\lambda_{j}^{t+1} &= \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{t} \\
\mu_{j}^{t+1} &= \frac{\sum_{i=1}^{n} p_{ij}^{t} x_{i}}{\sum_{i=1}^{n} p_{ij}^{t}} = \frac{\sum_{i=1}^{n} p_{ij}^{t} x_{i}}{n \lambda_{j}^{t+1}}. 
\end{align*}
\]

(1.17) (1.18)

Although this is called the maximization step, we see that no likelihood has been maximized. In fact, any weighted estimator for any parameter can be defined.

4. **Nonparametric Density Estimation Step:** For any real \( u \)

\[
\begin{align*}
f^{t+1}(u) &= \frac{1}{2nh} \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij}^{+} \left[ K\left( \frac{u - x_{i} + \mu_{j}^{t+1}}{h} \right) + K\left( \frac{-u - x_{i} + \mu_{j}^{t+1}}{h} \right) \right], 
\end{align*}
\]

(1.19)

where \( K(\cdot) \) is a kernel function and \( h \) is the bandwidth. This step produces a symmetric estimate of the density function.

We could skip the stochastic step of this algorithm and use the posterior probabilities \( p_{ij}^{t} \) instead of \( z_{ij}^{+} \) in the nonparametric density estimation step. Then equation (1.19) would be replaced by

\[
\begin{align*}
f^{t+1}(u) &= \frac{1}{2nh} \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij}^{t} \left[ K\left( \frac{u - x_{i} + \mu_{j}^{t+1}}{h} \right) + K\left( \frac{-u - x_{i} + \mu_{j}^{t+1}}{h} \right) \right]. 
\end{align*}
\]

(1.20)

We ran a simulation to compare the performance of the deterministic algorithm (spEM) with the stochastic version (spSEM). We generated \( n = 100 \) observations from a two-component normal mixture, with \( \lambda = 0.2, \mu_{1} = 1, \) and \( \mu_{2} = 4, \) and \( \sigma_{1} = \sigma_{2} = 1. \)
The comparison is based on 10,000 Monte Carlo replications in which the bandwidth \( h \) was selected according to the formula \( h = 0.9 \min(\hat{\sigma}_j, IQR_j/1.34)n^{-1/5} \), where \( \hat{\sigma}_j \) and \( IQR_j \) are, respectively, the sample standard deviation and the interquartile range for the \( j \)th component. Hence, in our example, since \( n = 100 \) and the component standard deviations are equal to 1, we have \( h \approx 0.358 \). We started both algorithms from the true parameter values, and we let the spEM algorithm run for only 20 iterations, relative to the 100 iterations allowed in the stochastic version, since the deterministic sequence of estimates requires fewer iterations to stabilize. Tables 1.1 and 1.2 show, respectively, the empirical mean squared error (MSE) and bias for \((\lambda, \mu_1, \mu_2, \sigma_1, \sigma_2)\) from the model (1.14) with \( f(\cdot) \) taken to be standard normal.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( \lambda )</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>( \sigma_1 )</th>
<th>( \sigma_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>spSEM</td>
<td>0.00324</td>
<td>0.2068</td>
<td>0.0274</td>
<td>0.0256</td>
<td>0.0165</td>
</tr>
<tr>
<td>spEM</td>
<td>0.00317</td>
<td>0.1738</td>
<td>0.0248</td>
<td>0.0177</td>
<td>0.0140</td>
</tr>
</tbody>
</table>

Table 1.1. MSE for the estimates for spSEM and spEM.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( \lambda )</th>
<th>( \mu_1 )</th>
<th>( \mu_2 )</th>
<th>( \sigma_1 )</th>
<th>( \sigma_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>spSEM</td>
<td>-0.01099</td>
<td>0.0183</td>
<td>-0.0503</td>
<td>0.0530</td>
<td>0.0501</td>
</tr>
<tr>
<td>spEM</td>
<td>-0.01096</td>
<td>0.0033</td>
<td>-0.0502</td>
<td>0.0327</td>
<td>0.0496</td>
</tr>
</tbody>
</table>

Table 1.2. Bias of the estimates for spSEM and spEM.

The bias and MSE for both spSEM and spEM algorithms are close. Empirically, we notice that the deterministic version is slightly more efficient than the stochastic version. Based on this, we will use the non-stochastic version when extending the semi-parametric EM algorithm to the multivariate case.
Chapter 2

EM-like Algorithm for Multivariate Mixtures

2.1 Introduction

In this chapter we present a refinement and generalization of the algorithm of Bordes et al. (2007). We extend the algorithm to work with multivariate finite mixtures, where the vectors are assumed to have conditionally independent coordinates. The density functions are completely unspecified, and it characterizes this algorithm as a fully nonparametric procedure. Since we do not have a likelihood that is to be maximized, the algorithm proposed is not an EM algorithm in the usual sense (Dempster et al., 1977). We keep the term “EM” associated with our algorithm, because it strongly resembles a true EM algorithm for the parametric mixture case.

The algorithm described here is similar to the one introduced in Benaglia et al. (2008). They differ by the fact that we adapt the previous algorithm to allow different bandwidths for each component and each block of coordinates. We compute the bandwidths using a generalization of Silverman’s rule of thumb (Silverman, 1986) for the mixture case. An example illustrating the improvement in the estimated component densities when assuming different bandwidths is shown.

We will start the chapter by describing the model and showing how to estimate the parameters. In Section 2.4 we discuss how to choose the smoothing parameters. We present in Section 2.5 some modifications to the model in order to incorporate some special cases, such as a location-scale model or a model with symmetric components. To conclude this chapter, we show an approach to select the number of components in the nonparametric finite mixture problem. The results obtained using this method are very satisfactory.
2.2 Nonparametric Multivariate Finite Mixture

Recall the model discussed in Section 1.1. Suppose the vectors $X_1, \ldots, X_n$ are a simple random sample from a finite mixture of $m > 1$ arbitrary distributions. The density of each $X_i$ may be written

$$g_\varphi(x_i) = \sum_{j=1}^{m} \lambda_j \phi_j(x_i), \quad \text{with} \quad \sum_{j=1}^{m} \lambda_j = 1, \quad x_i \in \mathbb{R}^r,$$

where $\lambda_j \geq 0$ are the mixing proportions, and $\varphi^t = (\lambda^t, \phi^t)^t = (\lambda_1, \ldots, \lambda_m, \phi_1, \ldots, \phi_m)$ denotes the parameter vector to be estimated. As we mentioned before, model (2.1) is not identifiable if no restrictions are placed on $\phi_1, \ldots, \phi_m$. We want to emphasize that, in this thesis, we do not consider that a permutation of the $m$ pairs $(\lambda_1, \phi_1), \ldots, (\lambda_m, \phi_m)$, known as label-switching, is considered an identifiability problem.

A common restriction placed on $\phi_1, \ldots, \phi_m$, necessary to make model (2.1) identifiable, which we adopt throughout this thesis, is that the coordinates of the $X_i$ vector are conditionally independent. Under this assumption, model (2.1) can be written as

$$g_\varphi(x_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}),$$

where $f_{jk}(\cdot)$ denotes a univariate density function and $\phi_j(\cdot) = \prod_{k=1}^{r} f_{jk}(\cdot)$. When $f_{jk}(\cdot)$ are assumed to come from a particular parametric family of densities, standard univariate mixture model techniques described in McLachlan and Peel (2000) or Titterington et al. (1985) may easily be extended to the multivariate case. However, the goal here is to avoid the parametric assumption. We introduce an algorithm for estimating the parameter vector $\varphi$ in model (2.2), where we do not assume that $f_{jk}(\cdot)$ comes from a family of densities that may be indexed by a finite-dimensional parameter vector.

Hall and Zhou (2003) have considered the model above with two mixture components ($m = 2$). They introduced a root-$n$ consistent nonparametric estimator for the density functions and the mixing proportion. They also proved that, when $r \geq 3$
and $m = 2$, the model (2.2) is nonparametrically identifiable under particularly mild regularity conditions.

Notice that when $f_{jk}$ does not depend on $k$, we have the case in which the $X_i$ are not only conditionally independent but identically distributed as well, and the model above becomes:

$$g_\Phi(x_i) = \sum_{j=1}^m \lambda_j \prod_{k=1}^r f_j(x_{ik}). \quad (2.3)$$

Several authors (Hettmansperger and Thomas, 2000; Cruz-Medina et al., 2004; Elmore et al., 2004) have considered the special case described by model (2.3). They developed the cutpoint approach we discussed in Section 1.2.

What distinguishes model (2.2) from model (2.3) is the assumption in the latter that $f_{j1}(\cdot) = \cdots = f_{jr}(\cdot)$ for all $j$. In some situations, however, this assumption may be too restrictive. At the same time, we may not wish to employ the fully general model because it might be reasonable to assume that some of the $f_{j1}(\cdot), \ldots, f_{jr}(\cdot)$ are the same. So, in order to encompass both the special case (2.3) and the more general case (2.2) simultaneously, we introduce the idea of grouping some of the coordinates into blocks, such that within a block, the coordinates are still conditionally independent but also identically distributed, and coordinates belonging to different blocks will have different density functions.

To write this model in a mathematical form, the notation will get a little more complicated, but we will try to make it clear. We will allow that the coordinates of $X_i$ are conditionally independent and there exist blocks of coordinates that are also identically distributed. These blocks may all be of size one so that case (2.2) is still covered, or there may exist only a single block of size $r$, which is case (2.3). If we let $b_k$ denote the block to which the $k$th coordinate belongs, where $1 \leq b_k \leq B$ and $B$ is the total number of such blocks, then equation (2.2) is replaced by

$$g_\Phi(x_i) = \sum_{j=1}^m \lambda_j \prod_{k=1}^r f_{jb_k}(x_{ik}). \quad (2.4)$$
The notation will remain consistent throughout this thesis and it is also consistent with Benaglia et al. (2008). In particular, we use the terms *component* and *coordinate* only to refer, respectively, to one of the distributions (subpopulations) making up the mixture and one of the measurements making up an observation vector. Table 2.1 shows the indices used, what each one denotes, and their range.

<table>
<thead>
<tr>
<th>Index</th>
<th>Denote</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>i</td>
<td>generic individual</td>
</tr>
<tr>
<td></td>
<td>j</td>
<td>component</td>
</tr>
<tr>
<td></td>
<td>k</td>
<td>coordinate</td>
</tr>
<tr>
<td></td>
<td>(\ell)</td>
<td>block</td>
</tr>
</tbody>
</table>

Table 2.1. Notation.

Note that \(m\), \(r\), and \(B\) stand for mixture components, repeated measurements, and blocks, and of course \(n\) has its usual meaning as the sample size.

To further elucidate model (2.4), consider an example where we have two groups of individuals and each individual in our sample is submitted to three different tests. Suppose there is reason to believe that the responses for tests one and three are similar, and the responses for test two differ from the others. Translating this to our model, we have a two component mixture with three repeated measures, where coordinates one and three will belong to one group, and we have a second group with the second coordinate. In our notation, \(1 \leq j \leq 2\), \(1 \leq k \leq 3\), \(b_1 = b_3 = 1\), \(b_2 = 2\), \(B = 2\), and for \(1 \leq i \leq n\),

\[
g_\phi(x_i) = \lambda f_{11}(x_{i1})f_{12}(x_{i2})f_{11}(x_{i3}) + (1 - \lambda)f_{21}(x_{i1})f_{22}(x_{i2})f_{21}(x_{i3}).
\]  

(2.5)

### 2.3 Nonparametric Estimation in Multivariate Mixtures

In this section we will present the algorithm to estimate the parameters in model (2.4). It is important to emphasize that we use the term EM in connection with this algorithm, although this is not an EM algorithm in the usual sense (Dempster et al., 1977) because there is no likelihood that this algorithm may be shown to maximize. However, we retain the name “EM” because, as in an EM algorithm for mixtures, we define \(Z_{ij} \in \{0, 1\}\) to be a Bernoulli random variable indicating that individual \(i\) comes from component \(j\). Since each individual comes from exactly one component, this implies
\[ \sum_{j=1}^{m} Z_{ij} = 1. \] In fact, we do not know from which component (group) each individual is from, so the \(Z_{ij}\)s are latent variables. Thus, the complete data is the set of all \((x_i, Z_i)\), \(1 \leq i \leq n\).

### 2.3.1 The Nonparametric EM (npEM) Algorithm

Suppose we are given initial values \(\varphi^0 = (\lambda^0, \ell^0)\). Then we iterate these three following steps until convergence of the \(\lambda\)s, that is, given a small \(\varepsilon > 0\), we stop the algorithm at the \((t + 1)\)th iteration when \(\max_{1 \leq j \leq m} |\lambda_{j}^{t+1} - \lambda_{j}^{t}| \leq \varepsilon\). Then, for \(t = 1, 2, \ldots\)

1. **E-step:** Calculate the posterior probabilities, conditional on the data and \(\varphi^t\), of component inclusion,

\[
p_{ij}^{t} \overset{\text{def}}{=} E_{\varphi^{t}}(Z_{ij} | x_i) = P_{\varphi^{t}}(Z_{ij} = 1 | x_i) = \frac{\lambda_{j}^{t} \prod_{k=1}^{r} f_{j b_{k}}^{t}(x_{ik})}{\sum_{j' = 1}^{m} \lambda_{j'}^{t} \prod_{k=1}^{r} f_{j'b_{k}}^{t}(x_{ik})} \tag{2.6}
\]

for all \(i = 1, \ldots, n\) and \(j = 1, \ldots, m\).

2. **M-step:** Set

\[
\lambda_{j}^{t+1} = \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{t} \tag{2.7}
\]

for \(j = 1, \ldots, m\).

3. **Nonparametric (Kernel) density estimation step:** For any real \(u\), define for each component \(j \in \{1, \ldots, m\}\) and each block \(\ell \in \{1, \ldots, B\}\)
\[ f_{j\ell}^{l+1}(u) = \frac{1}{h_{j\ell}} \sum_{k=1}^{r} \sum_{i=1}^{n} p_{ij}^{l} I\{b_{k} = \ell\} K\left( \frac{u - x_{ik}}{h_{j\ell}} \right) \]

\[ = \frac{1}{nh_{j\ell}C_{\ell}^{l+1}} \sum_{k=1}^{r} \sum_{i=1}^{n} p_{ij}^{l} I\{b_{k} = \ell\} K\left( \frac{u - x_{ik}}{h_{j\ell}} \right), \quad (2.8) \]

where \( K(\cdot) \) is a kernel density function, \( h_{j\ell} \) is the bandwidth for the \( j \)th component and \( \ell \)th block density estimate, and

\[ C_{\ell} = \sum_{k=1}^{r} I\{b_{k} = \ell\} \]

is the number of coordinates in the \( \ell \)th block. Considering the special cases, we notice that when the coordinates are not grouped into blocks, we have \( C_{\ell} = 1 \), and if we assume that the \( r \) coordinates are conditionally iid, then \( C_{\ell} = r \).

The bandwidths \( h_{j\ell} \) can either be chosen by the user or can be computed by a generalization of the usual rule of thumb used in kernel density estimation. We will show how to compute, and even iterate, the bandwidths in Section 2.4. Note that in Benaglia et al. (2008), \( h_{j\ell} \) is replaced by a single value of \( h \), for all \( 1 \leq j \leq m \), and \( 1 \leq \ell \leq B \).

In the case in which \( b_{k} = k \) for all \( k \), equation (2.8) becomes

\[ f_{jk}^{l+1}(u) = \frac{1}{nh_{jk}C_{\ell}^{l+1}} \sum_{i=1}^{n} p_{ij}^{l} K\left( \frac{u - x_{ik}}{h_{jk}} \right). \]

This algorithm is implemented in an R package (R Development Core Team, 2007) called mixtools (Young et al., 2005), available online from the Comprehensive R Archive Network (CRAN). In Appendix A we provide a short tutorial on how to use the functions that implement the methods described in this thesis.
To initialize the algorithm, it is often easier to start with an initial $n \times m$ matrix $\mathbf{P}^0 = (p^0_{ij})$ than with an initial parameter vector $\varphi^0$. Thus, during the first iteration, we skip directly to the M-step. To obtain this $\mathbf{P}^0$ matrix, we use a k-means clustering algorithm to assign each observation to one of the components, for which we only need to specify either the centers of the clusters or the total number of clusters (components). In this last case, the centers of the clusters are chosen at random. This procedure forces $\mathbf{P}^0$ to consist of just zeros and ones, and we find that it works well in practice. Surprisingly, the k-means can also be used for pure scale mixtures, which indicates that the npEM algorithm is robust with respect to the starting values.

There is a stochastic element in the original Bordes et al. (2007) algorithm. In the nonparametric density estimation step, each posterior probability $p_{ij}$ is replaced by $z^*_ij$ in equation (2.8), where $(z^*_i1, \ldots, z^*_im)$ is simulated from a multinomial random variable with a single trial and with probability vector given by $(p_{i1}, \ldots, p_{im})$. In various tests, we find consistent empirical evidence that the deterministic version is slightly, though not overwhelmingly, more efficient than the stochastic version. Because the deterministic algorithm does not require any additional overhead relative to the stochastic algorithm, we use it here exclusively; see the discussion around Table 1.1.

2.4 Bandwidth and Kernel Selection

Kernel density estimation is a well-studied topic in statistics (see Silverman, 1986; Scott, 1992; Härdle et al., 2004) and this is the technique we apply in the nonparametric density estimation step of our algorithm. Using the standardized posterior probabilities as the weights, we see that the mixture component densities are a weighted kernel density estimate. There are two things the user has to choose in this case: the kernel density $K(\cdot)$ and the bandwidth $h$, which is the smoothing parameter. The biggest issue is, in fact, the selection of the bandwidth, since this choice affects density estimates dramatically. With respect to the kernel density, much literature suggests that the choice of a kernel function does not have a dramatic impact on the resulting density estimate. Based on this, we simply take $K(t)$ to be the standard Gaussian density function.
Selecting a bandwidth is a more complicated issue. There are some standard rules in the literature to choose this smoothing parameter for the univariate case, when we do not have a mixture structure. We will use some of those ideas and extend them to the mixture case. We do not attempt a thorough exploration of this topic in this thesis, but we describe here some of our experience in choosing a bandwidth.

In Benaglia et al. (2008), where a single bandwidth is considered for all components and blocks densities, the entire $n \times r$ dataset is treated as a vector of length $nr$, and the default value for the bandwidth $h$ is set to be the rule of thumb due to Silverman (1986), page 48,

$$h = 0.9 \min \left\{ \text{SD}, \frac{\text{IQR}}{1.34} \right\} \left( nr \right)^{-1/5},$$

(2.10)

where SD and IQR are respectively the standard deviation and interquartile range of all $nr$ data values.

All rules used to select a bandwidth in the univariate density estimation problem are not appropriate in the nonparametric mixture setting, and it is not hard to see why they might produce an under- or over-estimate for the bandwidth. First, pooling all of the data implicitly treats all of the different components as though they are from the same distribution. This can overestimate the bandwidth, particularly if the mixture components’ centers are well-separated, because in that case, the variability of the pooled dataset will be larger than that of the individual components. Similarly, if the vector coordinates are not identically distributed within each component, the bandwidth could be biased upward for the same reason. In the underestimation side, notice that the expression $nr$ in equation (2.10) is an overestimate of the true sample size, especially when each of the $r$ coordinates gets a separate set of density estimates, in which case it may be sensible to eliminate the $r$ from the equation (2.10) entirely. But regardless if the coordinates are combined in blocks or not, there is also the fact that the “true” sample size from each component is actually some fraction of $n$, namely, about $\lambda_jn$ for the $j$th component.

Based on these arguments, we see that there is no reason to believe that the bandwidth should be the same for each component and for each block. Thus, we simply
modify the algorithm from Benaglia et al. (2008) by replacing \( h \) by \( h_{j\ell} \) in the nonparametric density estimation step as we have shown before, that is, we allow a different value of the bandwidth for each component and each block density function.

We chose to adapt Silverman’s rule of thumb (Silverman, 1986, page 48) to select the value of \( h_{j\ell} \) in an iterative procedure, in which, for each iteration of the \( \text{npEM} \) algorithm, we need information about the sample size, the sample standard deviation, and the interquartile range for each component and each block. In order to do this, we include in the M-step updates of the \( \mu_{j\ell} \) and \( \sigma_{j\ell} \) parameters for each \( 1 \leq j \leq m \) and \( 1 \leq \ell \leq B \):

\[
\hat{\mu}_{j\ell}^{t+1} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{r} \mathbb{I}_{\{b_k = \ell\}} x_{ik}}{\sum_{i=1}^{n} \sum_{k=1}^{r} \mathbb{I}_{\{b_k = \ell\}}} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{r} \mathbb{I}_{\{b_k = \ell\}} x_{ik}}{n \lambda_j^{t+1} C_{\ell}}
\]

(2.11)

\[
\hat{\sigma}_{j\ell}^{t+1} = \left[ \frac{1}{n C_{\ell} \lambda_j^{t+1}} \sum_{i=1}^{n} \sum_{k=1}^{r} \mathbb{I}_{\{b_k = \ell\}} (x_{ik} - \hat{\mu}_{j\ell}^{t+1})^2 \right]^{1/2}
\]

(2.12)

Therefore, the estimated bandwidths at the \( (t+1) \)th iteration is given by:

\[
\hat{h}_{j\ell}^{t+1} = 0.9 \min \left( \hat{\sigma}_{j\ell}^{t+1} \cdot \frac{\text{IQR}_{j\ell}}{1.34}, \left( \frac{1}{n \lambda_j^{t+1}} \right)^{-1/5} \right)
\]

(2.13)

where \( n \lambda_j^{t+1} \) estimates the sample size for the \( j \)th component, and \( \text{IQR}_{j\ell} \) is the weighted empirical interquartile range for the \( j \)th component and \( \ell \)th block. To compute \( \text{IQR}_{j\ell} \), we first find the cumulative sum of the posterior probabilities for the \( j \)th component associated with the vector of observations composed of the coordinates in the \( \ell \)th block, which we call weighted empirical cdf. Hence, \( \text{IQR}_{j\ell} \) is the difference between the 0.75 and the 0.25 quantiles of this weighted empirical cdf.
To illustrate the effect of having different bandwidths we will show a simple example. We generate 300 observations from the mixture:

\[
g(x_1, x_2, x_3) = 0.4N(0, 1)N(0, 1)N(0, 1) + 0.6N(15, 25)N(15, 25)N(15, 25). \tag{2.14}
\]

This example has the same structure as model (2.4), where \( m = 2, r = 3, \) and \( b_1 = b_2 = b_3 = 1, \) that is, we have a two-component mixture with three conditionally iid repeated measures. We estimate the model applying the npEM algorithm described in Subsection 2.3.1 using both a single bandwidth given by (2.10) and different bandwidths computed by (2.13).

If we consider each component separately and apply Silverman’s rule of thumb, the bandwidths would be \( h_1 = 0.9(0.4 \times 300)^{-1/5} \approx 0.35, \) and \( h_2 = 0.9 \times 5 \times (0.6 \times 300)^{-1/5} \approx 1.593, \) for the first and second component respectively. The npEM algorithm used \( \hat{h} = 1.923 \) for a single bandwidth, and \( \hat{h}_{11} = 0.274 \) and \( \hat{h}_{21} = 1.194 \) when allowing different bandwidths. We notice that \( \hat{h} \) overestimates the value of the bandwidth for the first component, which results in a density estimate with larger variance, as we can see in Figure 2.1 (left plot). The plot on the right in Figure 2.1 shows the component density estimates using \( \hat{h}_{11} \) and \( \hat{h}_{21} \), which are good estimates of the optimal bandwidths for each component, resulting in density estimates very close to the true component densities.

![Component Densities](image)

Figure 2.1. Effect of bandwidth in the component density estimates. On the left we have estimates with the same bandwidth, and on the right with different bandwidths.
2.5 Modifications to the General Model and Algorithm

The general model of equation (2.4) and the algorithm of Section 2.3.1 may be modified in various ways; see also Benaglia et al. (2008). It is hoped that imposing reasonable restrictions on the model will allow data to be combined and more efficient estimates will result.

2.5.1 Location-Scale Model

There are some plausible models that are more restrictive than (2.4) but not as restrictive as the case in which all coordinates are identically distributed. For instance, if in equation (2.4) we write \( \ell = b_k \), then

\[
f_{j\ell}(x) = \frac{1}{\sigma_{j\ell}} f_j \left( \frac{x - \mu_{j\ell}}{\sigma_{j\ell}} \right)
\]

for unknown parameters \((\mu_j, \sigma_j, f_j), j = 1, \ldots, m\). Note that \(\mu_j\) and \(\sigma_j\) are both \(B\)-vectors. With this restriction, the totally nonparametric specification of \(f_{j\ell}\) becomes a semiparametric specification. In this case, we assume that, depending on the individual’s mixture component, the coordinates within an individual have the same distribution shape, differing only in their location and scale parameters.

To implement the semiparametric EM algorithm in this case, equations (2.6) and (2.7) remain unchanged but it is necessary to modify equation (2.8) to account for the fact that all of the coordinates provide information about the form of each \(f_j\). Thus, in the case (2.15), equation (2.8) is replaced by

\[
f_{j}^{t+1}(u) = \frac{1}{n \rho h_j \lambda_{j}^{t+1}} \sum_{i=1}^{n} \sum_{k=1}^{r} \rho_{ij}^{t} K \left( \frac{u - (x_{ik} - \hat{\mu}_{jb_k})}{h_j \sigma_{jb_k}} \right).
\]
Furthermore, the M-step also includes updates of the $\mu_{j\ell}$ and $\sigma_{j\ell}$ parameters for each $1 \leq j \leq m$ and $1 \leq \ell \leq B$:

$$
\hat{\mu}_{j\ell}^{t+1} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\} x_{ik}}{\sum_{i=1}^{n} \sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\}} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\} x_{ik}}{n \lambda_{j}^{t+1} C_{\ell}}
$$

$$
\hat{\sigma}_{j\ell}^{t+1} = \left[ \frac{1}{n C_{\ell} \lambda_{j}^{t+1}} \sum_{i=1}^{n} \sum_{k=1}^{r} p_{ij}^{t} I\{b_k = \ell\} (x_{ik} - \mu_{j\ell}^{t+1})^2 \right]^{1/2}
$$

Naturally, it is possible to place constraints on the $\mu_j$ or $\sigma_j$ vectors when this is sensible. For instance, if the mixture is purely a location mixture, then we might stipulate that $\sigma_j = \sigma$ for each $j$ and for some $B$-vector $\sigma$. Similarly, we might stipulate that $\mu_j = \mu$ if the mixture is purely a scale mixture. In these latter two cases, note that we still allow the different blocks to have different scale and location parameters, though of course this may be restricted as well. Also note that because $f_j$ is completely unconstrained (except in special cases like Section 2.5.2), each element of the $\mu_j$ may only be identified up to a constant shift and each element of $\sigma_j$ may only be identified up to a constant multiple. Stated differently, there is no loss of generality in assuming that $\sum_{\ell} \mu_{j\ell} = 0$ and $\sum_{\ell} \sigma_{j\ell} = 1$ for each $j$; however, when implementing the algorithm, it is generally not necessary to enforce these constraints.

Now, if we assume that individual differences, i.e., the mixture components, only account for differences up to a location and scale parameter, but otherwise the distributions of different blocks of coordinates do not relate to one another in any way, the location-scale model has to be written as

$$
f_{j\ell}(x) = \frac{1}{\sigma_{j\ell}} f_{\ell} \left( \frac{x - \mu_{j\ell}}{\sigma_{j\ell}} \right)
$$

instead of equation (2.15). Notice that these two equations, which differ only in the replacement of a single $j$ by $\ell$, involve assumptions that are quite distinct.
Note also that the corresponding form of equation (2.8) looks quite different than its earlier counterpart in equation (2.16):

\[
f_{t+1}^{\ell}(u) = \frac{1}{nh_{t}C_{\ell}} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{r} I\{b_k = \ell\} p_{ij}^{t} K \left( \frac{u - (x_{ik} - \hat{\mu}_{j}^{t+1})}{h_{t} \sigma_{j}^{t+1}} \right). \tag{2.20}
\]

As an even more restrictive case of both (2.15) and (2.19), we may assume that all coordinates in all components have the same distributional shape, summarized by the density \( f(\cdot) \), and

\[
f_{j\ell}(x) = \frac{1}{\sigma_{j\ell}} f \left( \frac{x - \mu_{j\ell}}{\sigma_{j\ell}} \right). \tag{2.21}
\]

In case (2.21), equation (2.8) becomes

\[
f_{t+1}^{\ell}(u) = \frac{1}{nrh} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{r} p_{ij}^{t} K \left( \frac{u - (x_{ik} - \hat{\mu}_{j}^{t+1})}{h \hat{\sigma}_{j}^{t+1}} \right). \tag{2.22}
\]

Notice that we only have a single value of the bandwidth \( h \) due to the fact that we are assuming the same distributional shape for all observations, independent of their mixture component and coordinate. Although the default value of the bandwidth will be computed using the entire dataset as a single vector, we see that each observation will be previously standardized by its respective mean and standard deviation.

### 2.5.2 Symmetric Components

If we consider case (2.21) without repeated measures \( (r = 1) \) and for purely a location mixture \( (\sigma_{j} = 1, j = 1, \ldots, m) \), then the model becomes

\[
g_{\varphi}(x_i) = \sum_{j=1}^{m} \lambda_{j} f(x_i - \mu_{j}). \tag{2.23}
\]

When \( m = 2 \), equation (2.23) is exactly the location-shifted semiparametric mixture model that is proved identifiable by Bordes et al. (2006) and Hunter et al. (2007)
under the additional assumptions that $\lambda_1 \neq 0.5$ and that the density $f$ is symmetric about zero. This special case is also the model for which the original (semiparametric) stochastic EM algorithm is proposed in Bordes et al. (2007). In the non-stochastic version, equation (2.22) may be combined with a symmetrization step to give

$$f^{t+1}(u) = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{p_{ij}^t}{2nh} \left[ K\left( \frac{u - (x_i - \hat{\mu}_j^{t+1})}{h} \right) + K\left( \frac{-u - (x_i - \hat{\mu}_j^{t+1})}{h} \right) \right]. \quad (2.24)$$

### 2.5.3 Changing Block Structure

Model (2.4) allows us to have a different distribution for each component and for each block. Now, suppose that we want to assume that the block structure is different in each component. For instance, suppose that for one specific component, all the coordinates are conditionally independent and identically distributed, and for the other components we have blocks of identically distributed coordinates. The motivation to this structure came from the results we obtained when applying our algorithm to the water-level dataset (a dataset that will be discussed in Section 2.6).

In order to encompass this possibility, equation (2.4) has to be modified slightly to produce

$$g_{\varphi}(x_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jb_{jk}}(x_{ik}),$$

in which the only difference is that $b_k$ has been replaced by $b_{jk}$. This means that the block in which the $k$th observation falls depends on $j$ as well.

This generalization of the model is not currently implemented in the mixtools package, but it would be conceptually easy to do so. However, in this case, label-switching becomes a problem. By “label-switching”, we mean the result of permuting the labels of the $m$ components. When each component is assumed to follow the same model, it is not important which is labeled component 1, which is labeled component 2, etc. But if we now assume that, for example, component 1 has i.i.d. coordinates whereas components 2 and 3 have a different block structure, then it is necessary to ensure that “component 1” always refers to a particular one of the three components. This might be easiest to
achieve in practice using a two-step approach: First, obtain results for a model in which
the block structure is assumed the same for all three components. Then, use the final
posterior probabilities of component inclusion as starting values for a second algorithm
for fitting the more general model.

2.6 Selecting the Number of Components

So far in this thesis, we assumed that the number of components is known. This is
ture for examples where theory may suggest the number of components, but sometimes
it is unknown and it has to be determined. Our goal in this section is to show how
to choose the number of components that best fit the data. Because we have a fully
nonparametric approach, we are not able to use classical model selection methods such
as BIC and AIC, since the penalty used in these methods is, in general, an increasing
function of the number of parameters included in the fitted model. The approach we
consider here is the minimum distance method discussed by Chen and Kalbfleisch (1996)
in which, instead of linking the penalty to the number of parameters in the model, they
penalize the fit of mixtures with small mixing proportions, and consequently control the
number of components.

2.6.1 Kolmogorov-Smirnov Minimum Distance Method

The minimum distance method discussed by Chen and Kalbfleisch (1996) allows
us to work with most of the commonly used distances, such as Kolmogorov-Smirnov (KS),
Cramér-von Mises, Hellinger. Hettmansperger and Thomas (2000), for example, used
the Hellinger distance to select the number of components in an almost nonparametric
approach when the observations are vectors with conditionally iid coordinates. In our
case, we chose to use the Kolmogorov-Smirnov distance.

The basic idea is to find \( m \) that minimizes the penalized Kolmogorov-Smirnov
distance:

\[
D(\hat{F}_n(x), \hat{G}_m(x)) = \sup_{x \in \mathbb{R}^r} |\hat{F}_n(x) - \hat{G}_m(x)| - c_n \sum_{j=1}^m \log \hat{\lambda}_j,
\]  
(2.25)
where \( \hat{\lambda}_1, \ldots, \hat{\lambda}_m \) and \( \hat{G}_m(\cdot) \) are, respectively, the mixing proportions and cumulative distribution function (cdf) estimated by the nonparametric EM algorithm, \( c_n, n = 1, 2, \ldots \) is a sequence of positive constants, and \( \hat{F}_n(\cdot) \) is the empirical multivariate cdf computed by equation:

\[
\hat{F}_n(x^*) = \frac{1}{n} \sum_{i=1}^{n} I[x_{i1} \leq x^*_{1}, \ldots, x_{ir} \leq x^*_r] = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{r} I[x_{ij} \leq x^*_j],
\]

where \( I[.] \) is the indicator function. We compute \( \hat{G}_m(\cdot) \) using the formula:

\[
\hat{G}_m(x^*) = \sum_{j=1}^{m} \hat{\lambda}_j \prod_{k=1}^{r} \hat{F}_{jkb_k}(x^*_k),
\]

where \( \hat{F}_{jkb_k}(x^*_k) = \int_{-\infty}^{x^*_k} \hat{f}_{jkb_k}(x)dx \), that is, the estimated cdf for the \( j \)th component and \( b_k \)th block.

The penalty \( -\sum_{j=1}^{m} \log \hat{\lambda}_j \) is designed to control the number of components, by penalizing small values of \( \hat{\lambda}_j \)s, and to prevent fits with close subpopulations. Notice that when the number of components is unnecessarily large, some of the \( \hat{\lambda}_j \)s will be very small and this will make the penalty very large. Hence, this penalty controls the number of components by controlling the size of the \( \hat{\lambda}_j \)s. Suppose now that we have a true subpopulation with mixing proportion \( \lambda_1 \). If a fit introduces two or more subpopulations with mixing proportions, say \( \lambda_{11} \) and \( \lambda_{12} \), close to \( \lambda_1 \), the penalty for this part will be \( -c_n(\log \lambda_{11} + \log \lambda_{12}) \), which is much larger than \( -c_n \log \lambda_1 \). Thus, the penalty also prevents fits with close subpopulations.

The scaling constant \( c_n \) is selected so that the distance and penalty are of the same order of magnitude. Chen and Kalbfleisch (1996) applied \( c_n = 0.01 \frac{\log n}{\sqrt{n}} \) when using the Kolmogorov-Smirnov distance and, in fact, this constant worked well for our simulations. They say this choice of \( c_n \) is rather arbitrary, but we will show that it can be justified.
Denote $KS$ the usual Kolmogorov Smirnov distance. By the law of the iterated logarithm (see van der Vaart, 1998, page 268),

$$\limsup \sqrt{\frac{n}{2 \log \log n}} KS \leq \frac{1}{2}, \text{ almost surely.}$$

(2.28)

Hence,

$$KS \leq \frac{1}{2} \sqrt{\frac{2 \log \log n}{n}} = \sqrt{\frac{\log \log n}{2n}} \approx \frac{\log n}{5 \sqrt{2n}},$$

(2.29)

and then we will assume that $1/2$ of this upper limit is a common value for $KS$.

Now, considering that the mixing proportions are uniformly distributed, the penalty has a lower bound given by

$$- \sum_{j=1}^{m} \log \hat{\lambda}_j \geq - \sum_{j=1}^{m} \log \frac{1}{m} = -m \log \frac{1}{m}$$

(2.30)

Therefore, we want $c_n$ such that $KS \approx c_n \left(- \sum_{j=1}^{m} \log \hat{\lambda}_j \right)$ on average. Using the upper bound for $KS$ in (2.29), and the lower bound for the penalty in (2.30), we get

$$c_n \approx \frac{KS}{- \sum_{j=1}^{m} \log \lambda_j} \leq \frac{\log n}{-10 \sqrt{2n \log \frac{1}{m}}} = \frac{1}{10 \sqrt{2\log} \frac{1}{m}} \frac{\log n}{\sqrt{n}}.$$

(2.31)

Notice that there is a factor multiplying $\log n/\sqrt{n}$, which depends on the number of components. In Table 2.2 we calculated this value for different number of components.

<table>
<thead>
<tr>
<th>$m$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(10 \sqrt{2m \log \frac{1}{m}})^{-1}$</td>
<td>0.051</td>
<td>0.021</td>
<td>0.013</td>
<td>0.009</td>
<td>0.007</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Table 2.2. Values of factor in $c_n$ as a function of the number of components $m$

This table shows that the factor 0.01 selected by Chen and Kalbfleisch (1996) seems reasonable. But our calculations shows that it does not have to be exactly 0.01, it has to be a small coefficient. We tested different values and 0.01 worked well for our examples. Therefore, $c_n$ will be set as $c_n = 0.01 \frac{\log n}{\sqrt{n}}$ in our simulations as well.
A necessary condition to estimate the number of components consistently is identifiability. So, for a fixed number of repeated measures, we will consider only some of values of $m$ when minimizing (2.25), since the model is not identifiable for all $m = 1, 2, \ldots$. Identifiability is not fully settled in our case, but Hall et al. (2005) showed that, for a fixed number of components $m$, there is a inferior limit for the number of repeated measures, say $r_m$, so that the model could be identifiable. In other words, an $m$-component nonparametric mixture could be identifiable provided $r \geq r_m$, where $r_m \geq r_m^*$ and $2r_m^* - 1 \geq mr_m^* + 1$. It is important to emphasize that even when the number of measurements is larger than the inferior limit, there is no guarantee that the model is identifiable. In fact, all this rule says is that if the inferior limit is not reached, then the mixture is certainly not identifiable. On the other hand, in the conditionally iid case, Elmore and Wang (2003) show that a necessary and sufficient condition is to have $r$ repeated measures such that $r \geq 2m - 1$. Table 2.3 shows what we just discussed in numbers, that is, for a certain number of components, we see the minimum number of repeated measures necessary to guarantee identifiability. We stress that the number we see on the table for inferior limit given by Hall et al. (2005) is not a sufficient condition for identifiability.

<table>
<thead>
<tr>
<th>Number of components $m$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall et al. (2005)</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Elmore and Wang (2003)</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>13</td>
<td>15</td>
<td>17</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 2.3. Minimum number of repeated measures for identifiability.

Now, in order to select the number of components, we need to fix the range of $m$ for which the mixture is identifiable, and then apply the Kolmogorov-Smirnov minimum distance method. Then, for given a dataset, since we know how many repeated measures we have, we can use the conditions given in the previous paragraph and determine the maximum number components so the mixture is identifiable. Hettmansperger and Thomas (2000) wrote the condition given by Elmore and Wang (2003) in terms of $m$, so that $m \leq (r + 1)/2$. Because the most restrictive case in our model is when we assume
that the coordinates are conditionally iid, and the condition given by Elmore and Wang (2003) is necessary and sufficient for identifiability, we will be more conservative and assume this to be our limit as well. Hence, the Kolmogorov-Smirnov minimum distance method is summarized as:

$$m^* = \min_{1 \leq m \leq \frac{r+1}{2}} \{ D(F_n(x), \hat{G}_m(x)) \}$$ (2.32)

The next two subsections show how this method performs. First, we will generate data from $m = 2, 3, 4$ component multivariate normal mixture, and later we will apply the method to select the number of components in a real dataset.

### 2.6.2 Simulation Study

We generated $n$ observations from the multivariate normal mixture:

$$g(x_1, \ldots, x_r) = \sum_{j=1}^{m} \lambda_j N(\mu_j, \sigma_j I_r),$$ (2.33)

where $m$ is the number of components, $\mu_j$ is the component mean, $\sigma_j$ is the component standard deviation, and $I_r$ is the $r \times r$ identity matrix. Notice that we are considering conditionally iid repeated measures.

In fact, we will consider special cases of model (2.33) in our simulations. We will assume either a pure location or a pure scale mixture in our simulations. If we fix $\sigma_j = 1$ for all $j = 1, \ldots, m$, we have a pure location mixture. For a pure scale mixture, we fix $\mu_j = 0$ for all $j = 1, \ldots, m$. The details of each model we simulated are given below:

**Model 1.** We generated $n = 300$ observations with $r = 7$ iid repeated measures from two, three and four component location mixtures ($m = 2, 3, 4$), with $\mu_1 = 0, \mu_2 = 2, \mu_3 = 4, \mu_4 = 6$, and $\sigma_j = 1$ for all $j = 1, \ldots, m$.

**Model 2.** We generated $n = 1000$ observations with $r = 10$ iid repeated measures from two, three, and four component location mixtures ($m = 2, 3, 4$), with $\mu_1 = 0, \mu_2 = 2, \mu_3 = 4, \mu_4 = 6$, and $\sigma_j = 1$ for all $j = 1, \ldots, m$. 
Model 3. We generated \( n = 500 \) observations with \( r = 5 \) iid repeated measures from two, three, and four component scale mixtures \((m = 2, 3)\), with \( \mu_j = 0 \) for all \( j = 1, \ldots, m \), and \( \sigma_1 = 1, \sigma_2 = 10, \sigma_3 = 50 \).

For each case described above, we simulated 100 datasets and stored the number of components selected by the Kolmogorov-Smirnov minimum distance method.

<table>
<thead>
<tr>
<th>Model</th>
<th>( n )</th>
<th>( r )</th>
<th>Components ( 1 \leq m \leq 4 ) compared</th>
<th>True value of ( m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>300</td>
<td>7</td>
<td>1.00 1.00 0.81</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>10</td>
<td>1.00 1.00 0.70</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>5</td>
<td>0.99 0.66 —</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.4. Accuracy of the Kolmogorov-Smirnov Minimum Distance Method.

Table 2.4 presents the accuracy of the method, that is, the percentage of times that the method correctly selected the number of components, and we can see that the method is very accurate. For models 1 and 2, which are pure location models, the method precisely selected the right number of components 100\% of the time when the data was generated from two- and three-component mixtures. And for four components, the accuracy is 81\% in model 1 and 70\% in model 2, which is still high. For pure scale mixtures (model 3), the accuracy is 99\% and 66\%, respectively, for two and three components, and these are very good results since the nonparametric fitting of pure scale mixtures is a very complex problem. It is not easy to identify groups nonparametrically when they are centered at the same point.

To illustrate how the minimum distance method described in this section works, we selected one dataset generated from each model and the results are shown in the next three tables.

<table>
<thead>
<tr>
<th>Components</th>
<th>KS distance</th>
<th>penalty</th>
<th>( c_n \times ) penalty</th>
<th>penalized KS distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m = 1 )</td>
<td>0.23174</td>
<td>0</td>
<td>0</td>
<td>0.23174</td>
</tr>
<tr>
<td>( m = 2 )</td>
<td>0.04595</td>
<td>1.39215</td>
<td>0.00387</td>
<td>0.04982</td>
</tr>
<tr>
<td>( m = 3 )</td>
<td>0.04666</td>
<td>5.86297</td>
<td>0.01629</td>
<td>0.06295</td>
</tr>
</tbody>
</table>

Table 2.5. Selecting the number of components for a two component normal location mixture (Model 1 with \( m = 2 \)).
### Components

<table>
<thead>
<tr>
<th>Components</th>
<th>KS distance</th>
<th>penalty</th>
<th>$c_n \times$ penalty</th>
<th>penalized KS distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>0.1676</td>
<td>0</td>
<td>0</td>
<td>0.1676</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>0.07753</td>
<td>1.50611</td>
<td>0.00419</td>
<td>0.08171</td>
</tr>
<tr>
<td>$m = 3$</td>
<td><strong>0.02949</strong></td>
<td><strong>3.30213</strong></td>
<td><strong>0.00918</strong></td>
<td><strong>0.03866</strong></td>
</tr>
<tr>
<td>$m = 4$</td>
<td>0.07617</td>
<td>680.05078</td>
<td>1.89004</td>
<td>1.96621</td>
</tr>
</tbody>
</table>

Table 2.6. Selecting the number of components for a three component normal scale mixture (Model 2 with $m = 3$).

<table>
<thead>
<tr>
<th>Components</th>
<th>KS distance</th>
<th>penalty</th>
<th>$c_n \times$ penalty</th>
<th>penalized KS distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>0.49139</td>
<td>0</td>
<td>0</td>
<td>0.49139</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>0.18084</td>
<td>1.38648</td>
<td>0.00303</td>
<td>0.18387</td>
</tr>
<tr>
<td>$m = 3$</td>
<td>0.18665</td>
<td>3.47845</td>
<td>0.00760</td>
<td>0.19425</td>
</tr>
<tr>
<td>$m = 4$</td>
<td><strong>0.02801</strong></td>
<td><strong>5.54779</strong></td>
<td><strong>0.01212</strong></td>
<td><strong>0.04013</strong></td>
</tr>
<tr>
<td>$m = 5$</td>
<td>0.03039</td>
<td>9.10271</td>
<td>0.01988</td>
<td>0.05028</td>
</tr>
</tbody>
</table>

Table 2.7. Selecting the number of components for a four component normal location mixture (Model 3 with $m = 4$).

Table 2.5 presents the result for a dataset generated from model 1 with two components. Checking the Kolmogorov-Smirnov distances, we see that the values are very close for $m = 2$, and 3, but the method picks two components since the penalty is larger for three components. Table 2.6 is for a dataset representing model 2, and we see that three components leads to a better fit in this case. Table 2.7 contains the results for a dataset generated from model 3 with four components. Notice that the Kolmogorov-Smirnov distance is much smaller for $m = 4,5$ compared to $m = 1,2,3$, but because the method penalizes too many components, the right number of components is selected, which is four.

### 2.6.3 Choosing the Number of Components for the Water-level Task Data

The dataset we will use in this section, which it will be referred as the Waterdata, arises from the water-level task proposed by the Swiss psychologist Jean Piaget to assess children’s understanding of the physical world. The experiment is described in detail by Thomas et al. (1993), but in summary, it involves 405 children aged 11 to 16 years subjected to the water-level task. Each child is presented with a simple sketch of a
rectangular vessel with a cap, each tilted to one of the eight clock-hour orientations: 11, 4, 2, 7, 10, 5, 1, and 8 o’clock (in order that appears in the dataset). See Figure 2.2 for an illustration. The child is required to draw a line representing the surface of still liquid in the tilted vessel. Each vessel appeared with a horizontal reference line resembling a table surface. The response variable is the acute angle between the line drawn by the child and the horizontal line (correct orientation) defined as zero degrees, measured in degrees, with sign according to the slope of the line.

![Figure 2.2. Clock orientation and tilted vessel illustrating the Water-level task.](image)

We apply the minimum distance method to this dataset, allowing at most four components, since we have eight repeated measures (check Table 2.3). We have \( n = 405 \) observations, then the value of \( c_n \) is \( 0.01 \log(405)/\sqrt{405} = 0.00298 \). The npEM algorithm is used in three different settings, depending on the block structure of the coordinates, and the results are presented in the tables below. First, in Table 2.8, we assume the most general model, in which all the coordinates may have different distributions. Second, in Table 2.9, we assume four blocks with two coordinates in each, representing the clock-hour orientation pairs (1, 7), (2, 8), (4, 10), and (5, 11) o’clock. In the last one, Table 2.10, we assume that all eight coordinates are conditionally iid.

<table>
<thead>
<tr>
<th>Components</th>
<th>KS distance</th>
<th>penalty</th>
<th>( c_n \times ) penalty</th>
<th>penalized KS distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m = 1 )</td>
<td>0.16888</td>
<td>0</td>
<td>0</td>
<td>0.16888</td>
</tr>
<tr>
<td>( m = 2 )</td>
<td>0.15350</td>
<td>2.19621</td>
<td>0.00655</td>
<td>0.16005</td>
</tr>
<tr>
<td>( m = 3 )</td>
<td>0.08711</td>
<td>4.20302</td>
<td>0.01254</td>
<td>0.09965</td>
</tr>
<tr>
<td>( m = 4 )</td>
<td>0.08482</td>
<td>7.80360</td>
<td>0.02328</td>
<td>0.10810</td>
</tr>
</tbody>
</table>

Table 2.8. Selecting the number of components for the Waterdata when assuming non identically distributed coordinates.
Tables 2.8 and 2.9 selected three components, while Table 2.10 selected two components. However, in the iid case, we see that both the non-penalized and the penalized Kolmogorov-Smirnov distances for two and three components are very close. Based on these results, it seems that three components is appropriate for the water data.

### 2.6.4 Choosing the Number of Components for the RT Task Data

This dataset, also described in Miller et al. (2001) and Cruz-Medina et al. (2004), comes from a reaction time (RT) task, which is a common methodology used in developmental psychology. We will refer to this dataset as the RTdata. The experiment involves normally developing children 9 years old of age presented with two visual stimuli displayed on a computer monitor. The left image is the target stimulus, a two-dimensional line drawing of a simple shape, often resembling capital letters of the Roman alphabet, and on the right is either the identical stimulus or its mirror image. The child must press one key if the right stimulus matches the left or another key if the right stimulus is a mirror image. The child’s RT, in milliseconds, is recorded. Each child provides \( r = 6 \) responses to this task, and on each trial the target stimulus is a different shape.
The child could not anticipate what might appear on a subsequent trial because the six repeated measurements are embedded in a much longer sequence of trials for six other experimental conditions. This helped to assure independence of the repeated measures on each child, although we will discuss this issue in Chapter 3. The data consists of the RTs responses of 197 children who provided correct responses for all six task trials, i.e. who pressed the correct key.

We apply the minimum distance method here, allowing at most four components. We have $n = 197$ observations, then the value of $c_n$ is $0.01 \log(197)/\sqrt{197} = 0.00376$. We applied the npEM algorithm in two different settings. In the first one, we assumed that all the trials can have different distributions, and in the second one we assumed that the trials are conditionally iid. The results are presented in tables 2.11 and 5.3.

<table>
<thead>
<tr>
<th>Components</th>
<th>KS distance</th>
<th>penalty</th>
<th>$c_n \times$ penalty</th>
<th>penalized KS distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>0.22323</td>
<td>0</td>
<td>0</td>
<td>0.22323</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>0.11398</td>
<td>1.42633</td>
<td>0.00537</td>
<td>0.11935</td>
</tr>
<tr>
<td>$m = 3$</td>
<td><strong>0.10076</strong></td>
<td><strong>4.54496</strong></td>
<td><strong>0.01711</strong></td>
<td><strong>0.11786</strong></td>
</tr>
<tr>
<td>$m = 4$</td>
<td>0.09371</td>
<td>7.55080</td>
<td>0.02842</td>
<td>0.12214</td>
</tr>
</tbody>
</table>

Table 2.11. Selecting the number of components for the RT data when assuming non-identically distributed coordinates.

<table>
<thead>
<tr>
<th>Components</th>
<th>KS distance</th>
<th>penalty</th>
<th>$c_n \times$ penalty</th>
<th>penalized KS distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m = 1$</td>
<td>0.18905</td>
<td>0</td>
<td>0</td>
<td>0.18905</td>
</tr>
<tr>
<td>$m = 2$</td>
<td>0.09384</td>
<td>1.40375</td>
<td>0.00528</td>
<td>0.09912</td>
</tr>
<tr>
<td>$m = 3$</td>
<td><strong>0.08520</strong></td>
<td><strong>3.55908</strong></td>
<td><strong>0.01340</strong></td>
<td><strong>0.09860</strong></td>
</tr>
<tr>
<td>$m = 4$</td>
<td>0.08235</td>
<td>6.91512</td>
<td>0.02603</td>
<td>0.10838</td>
</tr>
</tbody>
</table>

Table 2.12. Selecting the number of components for the RT data when assuming conditionally iid coordinates.

For both settings, the method selected three components. We notice that, in both tables, the penalized Kolmogorov-Smirnov distances for two and three components are very close. Based on this, one can say that either two or three components would be more appropriate for this dataset.
Chapter 3

Monte Carlo Simulations

3.1 Introduction

In this chapter we will study the performance of the npEM algorithm proposed in Chapter 2. Using simulation techniques we will compare our results to the standard parametric normal procedure assuming conditionally independent coordinates. We will also make comparisons to some results existing in the literature when appropriate.

We tried to select a variety of models that not only represent possible situations in real life, but also models that allow us to explore the full capability of our algorithm. In this sense, we will simulate from pure location, pure scale, and asymmetric mixtures. We will have models with conditionally iid coordinates, models with blocks of conditionally iid coordinates, and also models where the coordinates come from very distinct density functions. For both the conditionally iid and the conditionally independent only cases, we chose some models specifically from examples used by other authors, so we can compare our method to those existing in the literature, such as the cut-point model (Elmore, 2003), the exponential tilted model (Qin and Leung, 2006; Wrobel, 2008), and the inversion method (Hall et al., 2005).

All the results presented in this chapter were obtained using the functions in the R package mixtools. Appendix A illustrates how to use the functions in this package that are relevant to this thesis.

3.2 Simulation Models and Results

In this section we will compare the performance of the npEM algorithm in eight different situations. Each model will be described individually and the estimates will be presented.
In every simulation, we started the npEM algorithm with an initial \( n \times m \) matrix of posterior probabilities \( \mathbf{P}^0 = (p_{ij}^0) \), which was determined by a k-means algorithm applied to each multivariate dataset, with initial cluster centers specified in each situation. By doing this, we prevent label switching.

The npEM algorithm results in estimates of the mixing proportions and the component density functions. Thus, we will evaluate the density estimates in terms of the Mean Integrated Squared Error (MISE). Suppose we ran \( S \) replications of a certain model, then the MISE is

\[
\text{MISE} = \frac{1}{S} \sum_{s=1}^{S} \int \left( \hat{f}_{jk}^{(s)}(u) - f_{jk}(u) \right)^2 du,
\]

for \( j = 1, \ldots, m \) and \( k = 1, \ldots, r \), where the integral is computed numerically. Each density \( \hat{f}_{jk}^{(s)} \) is computed by equation (2.8) using the final estimates of the posterior probabilities \( \hat{p}_{ij} \)'s, the mixing proportions \( \hat{\lambda}_j \)'s, and the bandwidths \( \hat{h}_{jk} \)'s in the \( s \)th replication.

Since basically any component feature can be estimated from its density, we will also report the means and variances of each component as an indicator of how well the algorithm performs.

which is a reasonable gold standard for comparison, since it is the parametric procedure that assumes the correct model.

In addition, we will also fit each mixture example using the parametric normal EM imposing conditionally independent coordinates. Notice that, when the data is in fact generated from normal mixtures, the normal EM is expected to perform better than any other method, since it assumes the correct model and it is likelihood based. In these situations, the normal EM is called the optimal procedure, since it is a reasonable gold standard for comparison. For these examples, our goal is to reach results close to the ones given by the parametric EM. In examples where the data does not come from normal mixtures, or at least some of the coordinates are not normally distributed, we expect to see a better performance of the npEM algorithm, since the normal EM would impose the wrong model.
Some of the models we simulate from have been studied by other authors as well. When this is the case, we will compare our results to the ones reported by these authors. For example, the first three models in this section will represent multivariate mixtures with conditionally iid repeated measures. For these models, we have the results from Elmore (2003), where the cut-point model is compared to the parametric normal procedure. The fourth model, a mixture of multivariate normal distributions with coordinates that are not identically distributed, has been studied by Qin and Leung (2006) using the exponential tilted method. Hall et al. (2005) introduced a method based on the inversion of the mixture and we will compare the performance of our algorithm with theirs in terms of the MISE.

3.2.1 Normal Location Mixture with Conditionally iid Coordinates

The first model we use is from a location mixture with normal component densities and conditionally iid coordinates. We write this model as

\[
g(x) = \lambda \prod_{k=1}^{8} \phi(x_k; 0, 1) + (1 - \lambda) \prod_{k=1}^{8} \phi(x_k; \mu, 1),
\]

where \( \phi(\cdot; \mu, 1) \) is the normal density with mean \( \mu \) and variance 1. Since the component standard deviations are both equal to one, the choice of \( \mu \) determines by how many standard deviations the component densities are separated. For example, if we take \( \mu = 2 \) then we are considering mixtures with component densities separated by two standard deviations.

We generated \( n = 100 \) observations from (3.2) with \( m = 2, r = 8, \lambda = 0.5 \) and \( \mu = 2 \). So we have a location mixture with normal components separated by two standard deviations. All the parameters were chosen to match exactly the example studied in Elmore (2003) so the results are comparable. Table 3.1 shows the results for \( \lambda, \mu_1, \sigma_1^2, \mu_2, \) and \( \sigma_2^2 \) considering three estimation methods: the npEM algorithm, the cut-point model, and the parametric normal EM with conditionally independent coordinates.

This is one of the situations where the normal EM is optimal, and therefore it is expected to perform better than the non- and semi-parametric procedures. In Table 3.1
we see that both the npEM algorithm and the cut-point model perform just as well as the optimal procedure in terms of estimating the parameters of this mixture.

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$ $\mu_1$ $\sigma_1^2$ $\mu_2$ $\sigma_2^2$</td>
</tr>
<tr>
<td>npEM</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
</tr>
<tr>
<td>Cut-point</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
</tr>
<tr>
<td>Normal</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
</tr>
</tbody>
</table>

Table 3.1. Two-component normal mixture with $\lambda = 0.5, \mu_1 = 0, \mu_2 = 2,$ and $\sigma_1^2 = \sigma_2^2 = 1$. The results are based on 1000 simulations with sample size $n = 100$ each.

Elmore (2003) also compared the cut-point model with the normal EM for this example, and he pointed out that the cut-point model seemed to be much more efficient with respect to estimating the component variances. However, we noticed that he misspecified the normal model by fitting a standard mixture of multivariate normal distributions with arbitrary variance-covariance matrices. The main assumption in our model is that the coordinates are independent conditional on the component membership, and so all the models we are comparing the npEM algorithm with have to carry the same assumption.

Figure 3.1 presents the estimated component densities and cdfs for an arbitrary replication of our simulation. The component MISEs are (0.00231, 0.00238) for the npEM algorithm, and (0.00066, 0.00070) for the normal procedure.

For the second example in the location-class mixture, we generated $n = 100$ observations from (3.2) with $m = 2, r = 8, \lambda = 0.25$ and $\mu = 1$. Here the two components are not “well-separated”.

Table 3.2 presents the results for $\lambda, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2$, considering the three estimation methods we mentioned in the first example. In this table we observe that even when the components are not very separated, the npEM algorithm and the cut-point
Figure 3.1. Estimated component density and cdf for a two-component normal mixture with $\lambda = 0.5$, $\mu_1 = 0$, $\mu_2 = 2$, and $\sigma_1^2 = \sigma_2^2 = 1$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

The model still performed as well as the normal procedure. The standard errors for the estimates are all of the same order. The bias for the mixing proportion $\lambda$ and component means $\mu_1$ and $\mu_2$ are smaller for the npEM algorithm. In terms of the component variances $\sigma_1^2$ and $\sigma_2^2$ the cut-point model gives estimates with smaller bias.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>$\lambda$</th>
<th>$\mu_1$</th>
<th>$\sigma_1^2$</th>
<th>$\mu_2$</th>
<th>$\sigma_2^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>npEM</td>
<td>Mean</td>
<td>0.25048</td>
<td>0.00013</td>
<td>1.01758</td>
<td>1.00110</td>
<td>0.99382</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
<td>0.06179</td>
<td>0.11090</td>
<td>0.12219</td>
<td>0.05394</td>
<td>0.06062</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.00048</td>
<td>0.00013</td>
<td>0.01758</td>
<td>0.00110</td>
<td>-0.00618</td>
</tr>
<tr>
<td>Cut-point</td>
<td>Mean</td>
<td>0.25950</td>
<td>0.00738</td>
<td>0.99959</td>
<td>1.00232</td>
<td>0.99454</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
<td>0.07176</td>
<td>0.12369</td>
<td>0.12394</td>
<td>0.05710</td>
<td>0.06626</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.00950</td>
<td>0.00738</td>
<td>-0.00041</td>
<td>0.00252</td>
<td>-0.00546</td>
</tr>
<tr>
<td>Normal</td>
<td>Mean</td>
<td>0.25207</td>
<td>-0.00409</td>
<td>0.99442</td>
<td>1.00377</td>
<td>0.99432</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
<td>0.06242</td>
<td>0.11898</td>
<td>0.12568</td>
<td>0.05666</td>
<td>0.06329</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.00207</td>
<td>-0.00409</td>
<td>-0.00558</td>
<td>0.00377</td>
<td>-0.00568</td>
</tr>
</tbody>
</table>

Table 3.2. Two-component normal mixture with $\lambda = 0.25$, $\mu_1 = 0$, $\mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 1$. The results are based on 1000 simulations with sample size $n = 100$ each.

Figure 3.2 shows the estimated component densities and cdfs. In this simulation, the component MISEs are (0.00518, 0.00193) for the npEM algorithm, and (0.00279,
0.00064) for the normal procedure. As in the previous example, the MISEs are larger for the nonparametric procedure when compared to the normal EM, but the estimated component pdfs and cdfs are still very close to the true functions.

![Graph showing estimated component density and cdf for a two-component normal mixture](image)

Figure 3.2. Estimated component density and cdf for a two-component normal mixture with $\lambda = 0.25, \mu_1 = 0, \mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 1$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

### 3.2.2 Laplace Location Mixture with Conditionally iid Coordinates

This model is still in the location-class mixtures and it is a two component mixture of Laplace (Double Exponential) distributions with conditionally iid coordinates. The model is represented by

$$g(x) = \lambda \prod_{k=1}^{8} \mathcal{L}(x_k; 0, 1) + (1 - \lambda) \prod_{k=1}^{8} \mathcal{L}(x_k; \mu, 1),$$  \hspace{1cm} (3.3)

where $\mathcal{L}(\cdot; \mu, \beta)$ is the Laplace density with mean $\mu$ and scale parameter $\beta$ expressed by

$$\mathcal{L}(x_k; \mu, \beta) = \frac{1}{2\beta} e^{-|x_k-\mu|/\beta},$$  \hspace{1cm} (3.4)

with $x_k \in \mathbb{R}$. The variance, under this parametrization, is $2\beta^2$. For our example, we will set $\beta = 1$, so the component variances are $\sigma_1^2 = \sigma_2^2 = 2$. 
We generated \( n = 100 \) observations from (3.2) with \( m = 2, r = 8, \lambda = 0.25 \) and \( \mu = 1 \), so the two components are very close together. Table 3.2 presents the results for \( \lambda, \mu_1, \sigma_1^2, \mu_2, \) and \( \sigma_2^2 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>( \lambda )</th>
<th>( \mu_1 )</th>
<th>( \sigma_1^2 )</th>
<th>( \mu_2 )</th>
<th>( \sigma_2^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>npEM</td>
<td>Mean</td>
<td>0.27193</td>
<td>0.03381</td>
<td>2.07990</td>
<td>1.00896</td>
<td>1.95796</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
<td>0.07995</td>
<td>0.22308</td>
<td>0.47448</td>
<td>0.10196</td>
<td>0.19025</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.02193</td>
<td>0.03381</td>
<td>0.07990</td>
<td>0.00896</td>
<td>-0.04204</td>
</tr>
<tr>
<td>Cut-point</td>
<td>Mean</td>
<td>0.25958</td>
<td>-0.00348</td>
<td>1.96639</td>
<td>1.00967</td>
<td>1.97900</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
<td>0.09970</td>
<td>0.20167</td>
<td>0.41106</td>
<td>0.09088</td>
<td>0.20959</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.00958</td>
<td>-0.00348</td>
<td>-0.03361</td>
<td>0.00967</td>
<td>-0.02100</td>
</tr>
<tr>
<td>Normal</td>
<td>Mean</td>
<td>0.31326</td>
<td>0.44078</td>
<td>2.29526</td>
<td>0.84752</td>
<td>2.00739</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
<td>0.10931</td>
<td>0.48929</td>
<td>1.33621</td>
<td>0.20984</td>
<td>0.60117</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
<td>0.06326</td>
<td>0.44078</td>
<td>0.29526</td>
<td>-0.15248</td>
<td>0.00739</td>
</tr>
</tbody>
</table>

Table 3.3. Two-component Laplace mixture with \( \lambda = 0.25, \mu_1 = 0, \mu_2 = 1, \) and \( \sigma_1^2 = \sigma_2^2 = 2 \). The results are based on 1000 simulations with sample size \( n = 100 \) each.

Notice that, in this particular example with mixture of Laplace distributions, the normal procedure assumes the wrong model. Hence, it is not a surprise that the semi- and non-parametric procedures do better. We see that the npEM algorithm estimates are very close to the cut-point model estimates and these two procedures lead to estimates with standard errors and bias much smaller than the normal EM. It is also necessary to emphasize that even when we say that the performance of the npEM algorithm is similar to the cut-point model, this is still an advantage for our algorithm, since we do not need to discretize the data, our approach if fully nonparametric, and we do not assume conditional iid components.

Another indicator of how well the npEM algorithm estimates the Laplace mixture is given in Figure 3.3, where we have the estimated component densities and cdfs for one random replication of our simulation. The component MISEs is \((0.01131, 0.00377)\) for the npEM algorithm and \((0.06705, 0.02636)\) for the normal procedure, which indicates the better performance of the nonparametric procedure. It is impressive how close the estimated curves are to the true component pdfs and cdfs, especially because the two components are basically blended together.
This example also illustrates the advantage of having a fully nonparametric procedure over the normal or any other parametric EM. Except for of course the right model, a parametric procedure would not be able to catch the shape of such component densities.

Figure 3.3. Estimated component density and cdf for a two-component Laplace mixture with $\lambda = 0.5, \mu_1 = 0, \mu_2 = 1$, and $\sigma_1^2 = \sigma_2^2 = 2$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

3.2.3 Normal Scale Mixture with Conditionally iid Coordinates

This model is from a scale-mixture class with normal component densities centered at zero and conditionally iid coordinates, that is,

$$g(x) = \lambda \prod_{k=1}^{8} \phi(x_k; 0, 1) + (1 - \lambda) \prod_{k=1}^{8} \phi(x_k; 0, \sigma^2),$$

where $\phi(\cdot; 0, \sigma^2)$ is the normal density with mean zero and variance $\sigma^2$. We will take $\sigma^2$ to be larger than one, so the second component will have variance $\sigma^2$ times larger than the first component density.

For the scale mixture example we generated $n = 100$ observations from (3.5) with $m = 2, r = 8, \lambda = 0.5$, component means equal to zero, $\sigma_1^2 = 1$, and $\sigma_2^2 = 9$. Table 3.4 shows the estimated parameters.
<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$</td>
</tr>
<tr>
<td>npEM</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
</tr>
<tr>
<td>Cut-point</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
</tr>
<tr>
<td>Normal</td>
<td>Mean</td>
</tr>
<tr>
<td></td>
<td>Std.Error</td>
</tr>
<tr>
<td></td>
<td>Bias</td>
</tr>
</tbody>
</table>

Table 3.4. Two-component normal mixture with $\lambda = 0.5, \mu_1 = \mu_2 = 0, \sigma^2_1 = 1,$ and $\sigma^2_2 = 9$. The results are based on 1000 simulations with sample size $n = 100$ each.

This is another example where the normal procedure is optimal. We can see in Table 3.4 that the npEM algorithm does a great job estimating the component means, but the bias of the estimated component variances are much larger than the other two procedures. In Figure 3.4 we have the estimated component pdfs and cdfs for one random replication of our simulation. The component MISEs are (0.00281, 0.00112) and (0.00078, 0.00022) for the nonparametric and normal procedures, respectively.

![Figure 3.4. Estimated component density and cdf for a two-component normal mixture with $\lambda = 0.5, \mu_1 = \mu_2 = 0, \sigma^2_1 = 1,$ and $\sigma^2_2 = 9$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.](image-url)
Pure scale mixtures are very difficult models to estimate, since there is no separation between components. Even so, the npEM is able to distinguish the two components in this example, resulting in estimates very close to the true pdfs and cdfs. Notice that we did not modify the algorithm to impose the scale structure, and we used the clustering k-means procedure to start the algorithm. Since the k-means split the dataset by location, the fact that still works for a purely scale mixture shows that the npEM algorithm is robust with respect to start values.

### 3.2.4 Normal Location Mixture with Coordinates not Identically Distributed

This is the first of the conditionally independent only models (not identically distributed). The model is written as

\[
g(x) = \lambda MN(x; \mu_1, \sigma_1^2 I_r) + (1 - \lambda)MN(x; \mu_2, \sigma_2^2 I_r),
\]

where \( MN(\cdot; \mu_j, \sigma_j^2 I_r) \) is the multivariate normal density with mean \( \mu_j \) and variance-covariance matrix \( \sigma_j^2 I_r \).

Here we generated \( n = 500 \) observations from model (3.6) with \( r = 3, \mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t, \) and \( \sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t \), for three values of the mixing proportion: \( \lambda = 0.3, 0.5, \) and \( 0.8 \). This example was also studied in Qin and Leung (2006), where they used an exponential tilt model to estimate the mixture. Since they showed the results in terms of the tilted parameters, we could not compare our results with theirs, but Wrobel (2008) used the same exponential tilt model and provided estimates for the component means and variances. We also estimated the parameters of the mixture using the parametric normal EM.

We replicated this simulation \( S = 500 \) times and the average of the estimates for each one of the parameters in the mixture with its respective standard error are presented in Table 3.5. In this table we have results for the npEM algorithm, the normal EM, and the exponential tilt model when \( \lambda = 0.3 \). We also have results for different values of the mixing proportion \( (\lambda = 0.5, 0.8) \), and the tables are in Appendix A.
The npEM algorithm performs just as well as the normal EM, and so does the exponential tilt model. The standard errors for the three procedures are all of the same order. Even with similar performance in terms of estimating the first two moments of each component density, the npEM algorithm has still the advantage of being a fully nonparametric approach, whereas the other two procedures are either parametric or semiparametric.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM Mean</th>
<th>npEM Std. Error</th>
<th>Normal Mean</th>
<th>Normal Std. Error</th>
<th>Exponential Tilt Mean</th>
<th>Exponential Tilt Std. Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.3</td>
<td>0.3014</td>
<td>0.0290</td>
<td>0.3017</td>
<td>0.0292</td>
<td>0.3010</td>
<td>0.0315</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0</td>
<td>-0.0038</td>
<td>0.0885</td>
<td>0.0022</td>
<td>0.0900</td>
<td>-0.0053</td>
<td>0.0998</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>0</td>
<td>-0.0070</td>
<td>0.1067</td>
<td>-0.0006</td>
<td>0.1087</td>
<td>-0.0200</td>
<td>0.1163</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0</td>
<td>0.0712</td>
<td>0.1237</td>
<td>0.0026</td>
<td>0.1165</td>
<td>0.0248</td>
<td>0.1390</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>1</td>
<td>0.9984</td>
<td>0.0550</td>
<td>0.9962</td>
<td>0.0543</td>
<td>1.0016</td>
<td>0.0571</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>1.5</td>
<td>1.4999</td>
<td>0.0611</td>
<td>1.4976</td>
<td>0.0609</td>
<td>1.5023</td>
<td>0.0640</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>2.5</td>
<td>2.4700</td>
<td>0.0678</td>
<td>2.5008</td>
<td>0.0681</td>
<td>2.4908</td>
<td>0.0737</td>
</tr>
<tr>
<td>$\sigma_{11}^2$</td>
<td>1</td>
<td>0.9931</td>
<td>0.1282</td>
<td>0.9912</td>
<td>0.1260</td>
<td>0.9940</td>
<td>0.1414</td>
</tr>
<tr>
<td>$\sigma_{12}^2$</td>
<td>1</td>
<td>0.9939</td>
<td>0.1374</td>
<td>0.9923</td>
<td>0.1382</td>
<td>0.9854</td>
<td>0.1559</td>
</tr>
<tr>
<td>$\sigma_{13}^2$</td>
<td>1</td>
<td>1.1680</td>
<td>0.2295</td>
<td>0.9930</td>
<td>0.1695</td>
<td>1.0552</td>
<td>0.2526</td>
</tr>
<tr>
<td>$\sigma_{21}^2$</td>
<td>1</td>
<td>0.9887</td>
<td>0.0768</td>
<td>0.9936</td>
<td>0.0774</td>
<td>0.9916</td>
<td>0.0780</td>
</tr>
<tr>
<td>$\sigma_{22}^2$</td>
<td>1</td>
<td>0.9838</td>
<td>0.0866</td>
<td>0.9910</td>
<td>0.0874</td>
<td>0.9937</td>
<td>0.0892</td>
</tr>
<tr>
<td>$\sigma_{23}^2$</td>
<td>1</td>
<td>1.0613</td>
<td>0.1142</td>
<td>0.9906</td>
<td>0.0964</td>
<td>1.0165</td>
<td>0.1244</td>
</tr>
</tbody>
</table>

Table 3.5. Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.3$. The results are based on 500 simulations with sample size $n = 500$ each.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\lambda$</th>
<th>Component 1 $f_{11}$</th>
<th>Component 2 $f_{21}$</th>
<th>Component 1 $f_{12}$</th>
<th>Component 2 $f_{22}$</th>
<th>Component 1 $f_{13}$</th>
<th>Component 2 $f_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>npEM</td>
<td>0.3</td>
<td>0.00526</td>
<td>0.00586</td>
<td>0.00671</td>
<td>0.00281</td>
<td>0.00586</td>
<td>0.00281</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.00382</td>
<td>0.00367</td>
<td>0.00389</td>
<td>0.00363</td>
<td>0.00367</td>
<td>0.00363</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.00241</td>
<td>0.00235</td>
<td>0.00267</td>
<td>0.00842</td>
<td>0.00235</td>
<td>0.00842</td>
</tr>
<tr>
<td>Normal</td>
<td>0.3</td>
<td>0.00207</td>
<td>0.00275</td>
<td>0.00345</td>
<td>0.00075</td>
<td>0.00275</td>
<td>0.00075</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.00129</td>
<td>0.00136</td>
<td>0.00172</td>
<td>0.00132</td>
<td>0.00136</td>
<td>0.00132</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.00073</td>
<td>0.00078</td>
<td>0.00094</td>
<td>0.00404</td>
<td>0.00078</td>
<td>0.00404</td>
</tr>
</tbody>
</table>

Table 3.6. MISE for the estimated density functions for each coordinate within each component in two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.3, 0.5, \text{ and } 0.8$. 
In Table 3.6 we computed the MISEs for both the npEM algorithm and the normal procedure for the different values of \( \lambda \), and Figure 3.5 presents the component pdfs and cdfs when \( \lambda = 0.3 \) for an arbitrary replication. We see that the estimates are very close to the true functions.

![Coordinate 1](image1)

![Coordinate 2](image2)

![Coordinate 3](image3)

Figure 3.5. Estimated component densities and cdfs for a two-component multivariate normal mixture with \( \mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1.5, 2.5)^t, \) and \( \sigma_1^2 = \sigma_2^2 = (1, 1)^t \) when \( \lambda = 0.3 \). The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

We ran a second simulation using (3.6), where the components are more separated. We considered the same sample size \( (n = 500) \), with parameters \( \mu_1 = (0, 0, 0)^t, \mu_2 = (2, 2.5, 3)^t, \) and \( \sigma_1^2 = (1, 1, 1)^t, \) and \( \sigma_2^2 = (1.5, 2, 1) \) for \( \lambda = 0.3, 0.5, \) and 0.8.

Table 3.7 shows the mean and standard errors of the estimates, when \( \lambda = 0.3 \), for the three methods: npEM algorithm, normal EM, and the exponential tilt model. The results when \( \lambda = 0.5, 0.8 \) are in Appendix A. What we see here is very similar to what we observed in the previous example. The npEM algorithm performs as well as
the optimal normal EM, and the standard errors for the estimates are also very close to
the ones obtained with the exponential tilt model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>npEM Mean</th>
<th>npEM Std.Error</th>
<th>Normal Mean</th>
<th>Normal Std.Error</th>
<th>Exponential Tilt Mean</th>
<th>Exponential Tilt Std.Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.3066</td>
<td>0.0221</td>
<td>0.3017</td>
<td>0.0217</td>
<td>0.3021</td>
<td>0.0232</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0.0140</td>
<td>0.0823</td>
<td>0.0031</td>
<td>0.0813</td>
<td>0.0023</td>
<td>0.0885</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>0.0181</td>
<td>0.0897</td>
<td>0.0036</td>
<td>0.0896</td>
<td>-0.0080</td>
<td>0.0917</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0.0540</td>
<td>0.0970</td>
<td>0.0043</td>
<td>0.0863</td>
<td>0.0212</td>
<td>0.0985</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>2.0049</td>
<td>0.0633</td>
<td>1.9957</td>
<td>0.0619</td>
<td>2.0031</td>
<td>0.0656</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>2.5</td>
<td>0.0779</td>
<td>2.4958</td>
<td>0.0771</td>
<td>2.5022</td>
<td>0.0814</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>3.0002</td>
<td>0.0557</td>
<td>3.0012</td>
<td>0.0556</td>
<td>2.9989</td>
<td>0.0590</td>
</tr>
<tr>
<td>$\sigma_{11}^2$</td>
<td>1</td>
<td>0.1055</td>
<td>0.9932</td>
<td>0.1218</td>
<td>1.0010</td>
<td>0.1338</td>
</tr>
<tr>
<td>$\sigma_{12}^2$</td>
<td>1</td>
<td>0.1398</td>
<td>0.9978</td>
<td>0.1288</td>
<td>1.0033</td>
<td>0.1367</td>
</tr>
<tr>
<td>$\sigma_{13}^2$</td>
<td>1</td>
<td>1.1349</td>
<td>0.9947</td>
<td>0.1386</td>
<td>1.0580</td>
<td>0.1976</td>
</tr>
<tr>
<td>$\sigma_{21}^2$</td>
<td>1.5</td>
<td>1.4756</td>
<td>1.4908</td>
<td>1.1117</td>
<td>1.4864</td>
<td>0.1155</td>
</tr>
<tr>
<td>$\sigma_{22}^2$</td>
<td>2</td>
<td>1.9615</td>
<td>1.9852</td>
<td>0.1644</td>
<td>1.9915</td>
<td>0.1600</td>
</tr>
<tr>
<td>$\sigma_{23}^2$</td>
<td>1</td>
<td>0.9949</td>
<td>0.9902</td>
<td>0.0787</td>
<td>0.9985</td>
<td>0.0837</td>
</tr>
</tbody>
</table>

Table 3.7. Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t$, $\mu_2 = (2, 2.5, 3)^t$, $\sigma_1^2 = (1, 1, 1)^t$, and $\sigma_2^2 = (1.5, 2, 1)$ when $\lambda = 0.3$. The results are based on 500 simulations with sample size $n = 500$ each.

In Table 3.8 we have the MISEs for both the npEM algorithm and the normal procedure for the different values of $\lambda$, and Figure 3.6 shows the component pdfs and cdfs when $\lambda = 0.3$ for an arbitrary replication.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\lambda$</th>
<th>Component 1</th>
<th>Component 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$f_{11}$</td>
<td>$f_{12}$</td>
</tr>
<tr>
<td>npEM</td>
<td>0.3</td>
<td>0.00483</td>
<td>0.00518</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.00358</td>
<td>0.00338</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.00233</td>
<td>0.00225</td>
</tr>
<tr>
<td>Normal</td>
<td>0.3</td>
<td>0.00177</td>
<td>0.00203</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.00117</td>
<td>0.00109</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.00066</td>
<td>0.00066</td>
</tr>
</tbody>
</table>

Table 3.8. MISE for the estimated density functions for each coordinate within each component in two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t$, $\mu_2 = (2, 2.5, 3)^t$, $\sigma_1^2 = (1, 1, 1)^t$, and $\sigma_2^2 = (1.5, 2, 1)$ when $\lambda = 0.3, 0.5$, and 0.8.
3.2.5 Mixture of Asymmetric Distributions

For this model we chose the gamma distribution to represent the class of asymmetric mixtures. A two component gamma mixture can be written as

$$g(x) = \lambda g(x_1; 2, 2)g(x_2; 2, 2)g(x_3; 2, 2) + (1 - \lambda)g(x_1; 5, 2)g(x_2; 10, 1)g(x_3; 15, 0.5),$$  

(3.7)

where $g(\cdot; \alpha, \beta)$ represents the gamma density function with shape parameter $\alpha$ and scale parameter $\beta$. Under this parametrization, the mean and variance is $\alpha\beta$ and $\alpha\beta^2$, respectively.

We generated $n = 300$ observations from model (3.7) with parameters $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2), \text{ and } \beta_2 = (2, 1, 0.5)$ for three different values of
the mixing proportion: $\lambda = 0.3, 0.5,$ and $0.8$. This simulation was replicated 1000 times. Table 3.9 shows the estimates (mean, standard error and bias) obtained when $\lambda = 0.5$, for the component means and variances in the mixture for both the npEM algorithm and the normal EM. The tables for $\lambda = 0.3, 0.8$ are in Appendix A.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM Mean</th>
<th>npEM Std.Error</th>
<th>npEM Bias</th>
<th>Normal Mean</th>
<th>Normal Std.Error</th>
<th>Normal Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.5</td>
<td>0.4795</td>
<td>0.0342</td>
<td>-0.0205</td>
<td>0.3962</td>
<td>0.0518</td>
<td>-0.1038</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>4</td>
<td>3.9125</td>
<td>0.2701</td>
<td>-0.0875</td>
<td>3.5047</td>
<td>0.3406</td>
<td>-0.4953</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>4</td>
<td>3.8912</td>
<td>0.2817</td>
<td>-0.1088</td>
<td>3.3964</td>
<td>0.3509</td>
<td>-0.6036</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>4</td>
<td>3.8846</td>
<td>0.2707</td>
<td>-0.1154</td>
<td>3.4889</td>
<td>0.5011</td>
<td>-0.5111</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>10</td>
<td>9.8314</td>
<td>0.3922</td>
<td>-0.1686</td>
<td>9.2934</td>
<td>0.5059</td>
<td>-0.7066</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>10</td>
<td>9.8509</td>
<td>0.3095</td>
<td>-0.1491</td>
<td>9.3636</td>
<td>0.4543</td>
<td>-0.6364</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>7.5</td>
<td>7.4535</td>
<td>0.1777</td>
<td>-0.0465</td>
<td>7.2031</td>
<td>0.2294</td>
<td>-0.2969</td>
</tr>
<tr>
<td>$\sigma^2_{11}$</td>
<td>8</td>
<td>7.5206</td>
<td>1.8108</td>
<td>-0.4794</td>
<td>4.4988</td>
<td>1.2003</td>
<td>-3.5012</td>
</tr>
<tr>
<td>$\sigma^2_{12}$</td>
<td>8</td>
<td>7.5118</td>
<td>1.7282</td>
<td>-0.4882</td>
<td>4.1472</td>
<td>1.1427</td>
<td>-3.8528</td>
</tr>
<tr>
<td>$\sigma^2_{13}$</td>
<td>8</td>
<td>7.5879</td>
<td>1.5684</td>
<td>-0.4121</td>
<td>5.1440</td>
<td>2.6178</td>
<td>-2.8560</td>
</tr>
<tr>
<td>$\sigma^2_{21}$</td>
<td>20</td>
<td>20.2587</td>
<td>3.1827</td>
<td>0.2587</td>
<td>21.6855</td>
<td>3.0520</td>
<td>1.6855</td>
</tr>
<tr>
<td>$\sigma^2_{22}$</td>
<td>10</td>
<td>10.6188</td>
<td>1.4724</td>
<td>0.6188</td>
<td>12.9130</td>
<td>1.9527</td>
<td>2.9130</td>
</tr>
<tr>
<td>$\sigma^2_{23}$</td>
<td>3.8</td>
<td>4.0417</td>
<td>0.6143</td>
<td>0.2917</td>
<td>5.7417</td>
<td>1.1172</td>
<td>1.9917</td>
</tr>
</tbody>
</table>

Table 3.9. Two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2), \text{ and } \beta_2 = (2, 1, 0.5)$ for $\lambda = 0.5$. The means and variances are $\alpha \beta$ and $\alpha \beta^2$, respectively. The results are based on 1000 simulations with sample size $n = 300$ each.

In this table we see that the npEM algorithm outperforms the parametric procedure. This is expected since the normal EM imposes a symmetric shape on the component densities, and the mixture we generate from is composed of asymmetric distributions. In our example, the first component is more asymmetric than the second, and in this case, the normal EM underestimates the component variances. The bias of the estimates are much smaller for the nonparametric procedure.

In Figure 3.7 we have the estimated densities and cdfs for an arbitrary replication of the simulation when $\lambda = 0.5$. The components are not too separated in this case, and we see that the algorithm is still capable of capturing the shape of each distribution. To see this fact in numbers, we also computed the MISEs for each coordinate within each component (see Table 3.10).
Figure 3.7. Estimated component densities and cdfs for a two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2),$ and $\beta_2 = (2, 1, 0.5)$ when $\lambda = 0.5$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\lambda$</th>
<th>$f_{11}$</th>
<th>$f_{12}$</th>
<th>$f_{13}$</th>
<th>$f_{21}$</th>
<th>$f_{22}$</th>
<th>$f_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>npEM</td>
<td>0.3</td>
<td>0.00428</td>
<td>0.00421</td>
<td>0.00425</td>
<td>0.00108</td>
<td>0.00136</td>
<td>0.00213</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.00300</td>
<td>0.00283</td>
<td>0.00287</td>
<td>0.00146</td>
<td>0.00201</td>
<td>0.00291</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.00215</td>
<td>0.00217</td>
<td>0.00214</td>
<td>0.00333</td>
<td>0.00469</td>
<td>0.00699</td>
</tr>
<tr>
<td>Normal</td>
<td>0.3</td>
<td>0.01457</td>
<td>0.01462</td>
<td>0.01767</td>
<td>0.00277</td>
<td>0.00232</td>
<td>0.00351</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01308</td>
<td>0.01312</td>
<td>0.01493</td>
<td>0.00259</td>
<td>0.00387</td>
<td>0.00780</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.01288</td>
<td>0.01313</td>
<td>0.01342</td>
<td>0.00584</td>
<td>0.01632</td>
<td>0.02855</td>
</tr>
</tbody>
</table>

Table 3.10. MISE for the estimated density functions for each coordinate within each component in a two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2),$ and $\beta_2 = (2, 1, 0.5)$ when $\lambda = 0.3, 0.5,$ and 0.8.
Notice that the MISEs are much smaller for the npEM procedure, and we can say that asymmetry does not affect the capability of the nonparametric algorithm in terms of estimating the shape of the component distributions.

### 3.2.6 Mixture of Mixed Distributions

In this model we chose a set of three different distributions to compose the mixture components, so that coordinates within a component would all come from distributions with different characteristics. Here is the model

\[
g(x) = \lambda \phi(x_1; 0, 1)g(x_2; 2, 2) \mathcal{L}(x_3; 0, 4) + (1 - \lambda)g(x_1; 5, 0.5) \mathcal{L}(x_2; 0, 1) \phi(x_3; 10, 4),
\]

where \( \phi(\cdot; \mu, \sigma^2) \) represents the normal density function with mean \( \mu \) and variance \( \sigma^2 \), \( g(\cdot; \alpha, \beta) \) is the gamma density with parameters \( \alpha \) and \( \beta \), and \( \mathcal{L}(\cdot; \mu, \beta) \) denotes the Laplace density with mean \( \mu \) and scale parameter \( \beta \).

We generated \( n = 300 \) from the model above for \( \lambda = 0.3, 0.5, \) and 0.8. We computed the mean and variance for each density in the mixture, and the estimates for these parameters are given in Table 3.11 when \( \lambda = 0.5 \), for both the npEM and the normal procedures. The tables for the simulations with the other values of \( \lambda \) are in Appendix A. Table 3.12 presents the MISEs for the two procedures we are comparing for three different values of the mixing proportion \( \lambda \).

We want to emphasize that the npEM algorithm has as a goal the estimation of the mixing proportions and the component densities. We have been showing the estimates of the parameter in the mixtures, but especially in this example, it is more interesting to compare the MISEs, since the results in Table 3.11 do not contain much information about the difference in performance of the two procedures. Notice that the MISE is smaller in the normal EM only when the coordinates are in fact normal distributed, that is, the first coordinate in the first component \( (f_{11}) \), and the third coordinate in the second component \( (f_{23}) \). For all the other component densities, we see that the npEM performs much better than the parametric EM.
The estimated pdfs and cdfs for an arbitrary replication are shown in Figure 3.8. Looking at these plots for each coordinate, we see that different coordinates come from different distributions, and it is clear how efficient the npEM algorithm is for estimating each component distribution for each one of the three coordinates.
Consider the mixture

\[
g(\mathbf{x}) = \lambda_1 \prod_{k=1}^{3} \phi(x_k; 0,1) \prod_{k=4}^{5} \phi(x_k; 0,2) + \lambda_2 \prod_{k=1}^{3} \phi(x_k; 2.5, 1) \prod_{k=4}^{5} g(x_k; 10, 0.5) + \\
+ \lambda_3 \prod_{k=1}^{3} \phi(x_k; 5, 1) \prod_{k=4}^{5} \mathcal{L}(x_k; 10, 1),
\]

(3.9)
where \( \phi(\cdot; \mu, \sigma^2) \) denotes the normal density function with mean \( \mu \) and variance \( \sigma^2 \),
g\( (\cdot; \alpha, \beta) \) is the gamma density with parameters \( \alpha \) and \( \beta \), and \( \mathcal{L}(\cdot; \mu, \beta) \) is the Laplace density with mean \( \mu \) and scale parameter \( \beta \). The first three coordinates belong to the first block, while the second block is formed by the last two coordinates.

It is important to emphasize that none of the methods existing in the literature to estimate multivariate mixtures with conditionally iid coordinates are able to estimate this model assuming the blocking structure. They would either assume that all the coordinates are conditionally iid or they all have different distributions.

The estimates for the component means and variances within each block are given in Table 3.13 for the npEM and the normal procedures. We also report the standard errors and the bias.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( \lambda_1 )</th>
<th>( \lambda_2 )</th>
<th>( \lambda_3 )</th>
<th>( \mu_{11} )</th>
<th>( \mu_{12} )</th>
<th>( \mu_{21} )</th>
<th>( \mu_{22} )</th>
<th>( \mu_{31} )</th>
<th>( \mu_{32} )</th>
<th>( \sigma_{11}^2 )</th>
<th>( \sigma_{12}^2 )</th>
<th>( \sigma_{21}^2 )</th>
<th>( \sigma_{22}^2 )</th>
<th>( \sigma_{31}^2 )</th>
<th>( \sigma_{32}^2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>0.25</td>
<td>0.35</td>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
<td>2.50</td>
<td>5.00</td>
<td>5.00</td>
<td>10.00</td>
<td>1.00</td>
<td>4.00</td>
<td>1.00</td>
<td>2.50</td>
<td>1.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Mean</td>
<td>0.2504</td>
<td>0.3506</td>
<td>0.3990</td>
<td>-0.0063</td>
<td>-0.0024</td>
<td>4.9974</td>
<td>4.9976</td>
<td>4.9971</td>
<td>9.9948</td>
<td>0.9558</td>
<td>3.9784</td>
<td>0.9980</td>
<td>2.4936</td>
<td>0.9971</td>
<td>1.9871</td>
</tr>
<tr>
<td>Std.Error</td>
<td>0.0244</td>
<td>0.0276</td>
<td>0.0280</td>
<td>0.0670</td>
<td>0.1661</td>
<td>0.0566</td>
<td>0.0532</td>
<td>0.0937</td>
<td>0.0970</td>
<td>0.4647</td>
<td>-0.0216</td>
<td>0.0791</td>
<td>0.2804</td>
<td>0.0751</td>
<td>0.2887</td>
</tr>
<tr>
<td>Bias</td>
<td>0.0004</td>
<td>0.0006</td>
<td>-0.0010</td>
<td>-0.0063</td>
<td>-0.0024</td>
<td>-0.0002</td>
<td>-0.0024</td>
<td>-0.0052</td>
<td>-0.0070</td>
<td>-0.0216</td>
<td>0.0021</td>
<td>-0.0020</td>
<td>-0.0064</td>
<td>-0.0029</td>
<td>-0.0129</td>
</tr>
</tbody>
</table>

Table 3.13. Three-component mixture with two blocks of conditionally iid coordinates and \( \lambda = (0.25, 0.35, 0.4) \). The results are based on 1000 simulations with sample size \( n = 300 \) each.

In Table 3.14 we have the MISEs for the estimated densities for both the npEM and the normal procedures, and in Figure 3.9 we see the estimated pdfs and cdfs for each block of iid coordinates within each component.
Table 3.14. MISE for the estimated density functions for a three component mixture with two blocks of conditionally iid coordinates and $\lambda = (0.25, 0.35, 0.4)$.

<table>
<thead>
<tr>
<th>Method</th>
<th>Block</th>
<th>Component 1</th>
<th>Component 2</th>
<th>Component 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>npEM</td>
<td>1</td>
<td>0.00353</td>
<td>0.00263</td>
<td>0.00250</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.00239</td>
<td>0.00255</td>
<td>0.00494</td>
</tr>
<tr>
<td>Normal</td>
<td>1</td>
<td>0.00116</td>
<td>0.00081</td>
<td>0.00071</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.00086</td>
<td>0.00479</td>
<td>0.02183</td>
</tr>
</tbody>
</table>

Figure 3.9. Estimated component densities and cdfs for a three-component mixture with two blocks of conditionally iid coordinates and $\lambda = (0.25, 0.35, 0.4)$. The solid line represents the estimates by the npEM algorithm and the dashed black line represents the true functions.
We see that the estimates are all very close to the true functions. The component distributions we chose are not very asymmetric. In fact, they can be well approximated by a normal distribution, except for the Laplace density in the third component and second block of coordinates. This explains the low MISEs for the normal EM. However, notice that the MISE for the npEM algorithm associated to the block of Laplace distributions are much smaller than the MISE obtained by the parametric procedure.

The advantage of the blocking structure is that, if two coordinates are conditionally iid and we group them into a block, the estimates will be more consistent than if we just estimate one single density for each coordinate, because we will have more information for that specific distribution.

### 3.2.8 Comparing the npEM with the inversion method of Hall et al. (2005)

In this subsection we will compare the performance of the npEM algorithm to the estimation technique presented by Hall et al. (2005), which is a method based on the inversion of the mixture model. We will use the same synthetic examples for which Hall et al. (2005) tested their methodology. The two methods will be compared in terms of the MISE defined in (3.1).

The three simulated models, described below, are trivariate two-component mixtures \((m = 2, r = 3)\) with independent repeated measures.

1. The first model is a normal mixture written as

   \[
g(x) = \lambda MN(x; \mu_1, I_3) + (1-\lambda) MN(x; \mu_2, I_3),
   \]

   where \(MN(\cdot; \mu, I_3)\) denotes the trivariate normal density with mean \(\mu\) and variance-covariance matrix \(3 \times 3\). The component means are \(\mu_1 = (0, 0, 0)^t\) and \(\mu_2 = (3, 4, 5)^t\).

2. The second model uses Laplace distributions given by

   \[
g(x) = \lambda \prod_{k=1}^3 \mathcal{L}(x_k; 0, 1) + (1-\lambda) \prod_{k=1}^3 \mathcal{L}(x_1; 3, 1),
   \]
where $\mathcal{L}(\cdot; \mu, \beta)$ is the Laplace pdf with mean $\mu$ and scale parameter $\beta$. The component means are $\mu_1 = (0, 0, 0)^t$ and $\mu_2 = (3, 3, 3)^t$.

3. In the third model we have central and non-central $t$ distributions, that is,

$$g(x) = \lambda t_{(10)}(x_1)t_{(10)}(x_2)t_{(10)}(x_3) + (1 - \lambda)t_{(10)}(x_1; 3)t_{(10)}(x_1; 4)t_{(10)}(x_1; 5),$$

where $t_{(10)}$ is the central $t$ distribution with ten degrees of freedom, and $t_{(10)}(\cdot; \delta)$ denotes the non-central $t$ distribution with ten degrees of freedom and non-centrality parameter $\delta$. The first component has non-centrality parameter equal to zero and thus the component mean is $\mu_1 = (0, 0, 0)^t$, whereas the second component’s coordinates have non-centrality parameters 3, 4, and 5, respectively. Thus, the component mean is $\mu_2 = 1.0837 \times (3, 4, 5)^t$.

Recall that in all three examples the coordinates are independently distributed conditional on their component membership. It is necessary to emphasize that, even though the true model in the Laplace example has conditionally iid coordinates, the npEM algorithm used in this simulation assumes the general model (2.2), where each coordinates has its own distribution.

We generated $n = 500$ observations from each model and the simulation was replicated $S = 300$ times for values of $\lambda = 0.1, 0.2, 0.3,$ and $0.4$. We started each algorithm with an initial $n \times m$ matrix of posterior probabilities $P^0 = (p^0_{ij})$, which was determined by a k-means algorithm applied to each trivariate dataset, with initial cluster centers $(0, 0, 0)$ and $(4, 4, 4)$. This procedure was used here to prevent label-switching among replications, since we are simulating from location-shifted models. In comparison, Hall et al. (2005) dealt with label-switching by enforcing the constraint $\hat{\lambda}_1 < \hat{\lambda}_2$.

A comparison between the npEM algorithm and the inversion method was previously presented in Benaglia et al. (2008), but the difference is that we used the npEM algorithm with different bandwidths, whereas Benaglia et al. (2008) used a single bandwidth, which was computed as the default value (Silverman’s rule of thumb). Talking about tuning parameters such as the bandwidth, though the inversion method has several
other tuning parameters, Hall et al. (2005) used near-optimal values derived by fitting the models with the normal procedure.

Figure 3.10 shows the square roots of the MISEs on a logarithmic scale as a function of $\lambda$, the mixing proportion for the first component, for each model described above for all component densities $f_{jk}$, $j = 1, 2$ and $k = 1, 2, 3$. The results for the inversion algorithm of Hall et al. (2005) are approximated from their plots on page 675, Fig. 2.

In this figure we see that the npEM algorithm dramatically outperforms the inversion method for the three models. Notice that the horizontal dotted line at $\sqrt{\text{MISE}} = 0.15$ separates the two sets of results entirely, which means that the smallest value of MISE for the inversion method for any example is greater than the greatest value of MISE for our npEM algorithm. Similar results were reported in Benaglia et al. (2008), but the MISEs are a little bit smaller in our case, where we used the npEM algorithm allowing different bandwidths.

![Figure 3.10](image)

Figure 3.10. Square roots of MISEs on a logarithmic scale as a function of $\lambda$, for each model for all $f_{jk}$, $j = 1, 2$ and $k = 1, 2, 3$.

In each model, the three coordinates within a component are very similar or even identical. Thus, we would expect to see the lines representing the $\sqrt{\text{MISE}}$ for the coordinate densities to be very close to each other. In Figure 3.10, for the npEM
algorithm, we do not distinguish among them, whereas for the inversion technique, the lines are very distinct. This fact shows that the npEM algorithm is much more stable than the inversion method of Hall et al. (2005) in terms of estimating the component densities.

In terms of the mixing proportions, when $\lambda$ is small it means that a large proportion of the sample gives information about the second component, and thus it is not a surprise that we observe MISEs much smaller for the second component than the first when $\lambda$ is small. However, the values appear to converge as $\lambda$ nears 1/2.

### 3.3 Discussion

In this chapter we presented several examples to evaluate the performance of the npEM algorithm. In summary, the npEM algorithm performed really well in every example we showed. When the data was simulated from a mixture of normal distributions, the npEM performed as well as the normal EM, which is the optimal procedure in this situation. For the mixture of Laplace distributions with conditionally iid coordinates in 3.2.2, we saw that the nonparametric procedure produced much better results than the normal EM, which is not a surprise since the parametric procedure assumes the wrong model. The impressive fact about this example was how well the algorithm estimated the component cdfs and pdfs, especially because the two components were essentially blended together.

In comparison to semiparametric procedures such as the cut-point model (Elmore, 2003), and the exponential tilt EM (Qin and Leung, 2006, Wrobel, 2008), the npEM presented similar results. However, there is still an advantage for our algorithm, since we do not need to assume conditionally iid coordinates as in the cut-point model; our algorithm is computationally simpler than the exponential tilt EM; and our approach is fully nonparametric.

In our last simulation, we compared the npEM algorithm with the inversion method of (Hall et al., 2005). The results were presented in terms of the MISE of the estimates, and our algorithm dramatically outperforms the inversion method.
Chapter 4

Checking the Conditional Independence Assumption

4.1 Introduction

All the models presented in Chapter 2 assume that the coordinates are conditionally independent, that is, they are independent conditional upon component membership. This is a critical assumption in the development of those models. The goal in this chapter is to present some tools that will help to identify when it is reasonable to believe that conditional independence is satisfied. It is important to emphasize that these tools will actually be testing if the coordinates are conditionally uncorrelated, which does not necessarily imply conditional independence. However, they can still be used, because if the coordinates are conditionally correlated, then we know that they cannot be conditionally independent.

In order to develop these tools, we need to study the correlation in mixture models. Let $X_1, \ldots, X_n$ be iid observations from a two-component multivariate mixture:

$$g(x_i) = \lambda f_1(x_i) + (1 - \lambda)f_2(x_i),$$

where $x_i \in \mathbb{R}^r$, and $f_j(\cdot)$ is the $j$th component multivariate density function. Notice that we are not assuming conditionally independent coordinates. We will work with a two-component mixture in this section just to make it simpler, but the results can be easily extended to the $m$-components case, with $m > 2$.

In this chapter, we will discuss the unconditional and conditional correlation in mixture models of the form (4.1). We will see that the mixture structure itself imposes some correlation in the data, which makes it more difficult to assess information about the unconditional correlation. We need to be able to distinguish if the source of correlation is
solely from the mixing structure or if it is from the mixing structure along with correlation within components.

4.2 Deriving the Correlation in Finite Mixture Models

Denote $X_k$ to be the $k$th coordinate of the vector $X$. Let us also keep in mind the following notation:

- $\mu_{jk}$ is the mean for the $k$th coordinate in the $j$th component,
- $\sigma^2_{jk}$ is the variance for the $k$th coordinate in the $j$th component, and
- $\sigma^{(j)}_{kk'} = Cov(X_k, X_{k'}|j)$, that is, the conditional covariance between $X_k$ and $X_{k'}$ given the component $j = 1, 2$, for $1 \leq k, k' \leq r$ and $k \neq k'$. Under the assumption of conditional independence, we have $\sigma^{(j)}_{kk'} = 0$, for all $j$.

The unconditional expectation and variance of $X_k$ is:

$$E(X_k) = \lambda \mu_{1k} + (1 - \lambda) \mu_{2k}$$

$$Var(X_k) = \lambda \sigma^2_{1k} + (1 - \lambda) \sigma^2_{2k} + \lambda(1 - \lambda)(\mu_{1k} - \mu_{2k})^2$$

In order to find the unconditional covariance between $X_k$ and $X_{k'}$, we need:

$$E(X_k X_{k'}) = \lambda \left( \sigma^{(1)}_{kk'} + \mu_{1k} \mu_{1k'} \right) + (1 - \lambda) \left( \sigma^{(2)}_{kk'} + \mu_{2k} \mu_{2k'} \right)$$

Hence, the unconditional $Cov(X_k, X_{k'})$ is given by:

$$Cov(X_k, X_{k'}) = E(X_k X_{k'}) - E(X_k) E(X_{k'})$$

$$= \lambda \sigma^{(1)}_{kk'} + (1 - \lambda) \sigma^{(2)}_{kk'} + \lambda(1 - \lambda)(\mu_{1k} - \mu_{2k})(\mu_{1k'} - \mu_{2k'})$$

Notice that when the coordinates are conditionally independent, the unconditional means and variances remain the same, but the expectation of the cross products given
by (4.4) will be replaced by $\mathbb{E}(X_k X_{k'}) = \lambda \mu_{1k} \mu_{1k'} + (1 - \lambda) \mu_{2k} \mu_{2k'}$, which leads to:

$$\text{Cov}(X_k, X_{k'}) = \lambda(1 - \lambda)(\mu_{1k} - \mu_{2k})(\mu_{1k'} - \mu_{2k'})$$  \hspace{1cm} (4.6)

We might also point out that if the model is a pure scale mixture with conditionally independent coordinates, we have $\text{Cov}(X_k, X_{k'}) = 0$. This is because, for each coordinate, the component means are the same and then the difference between means in (4.6) is zero.

Now that we have the unconditional covariance, we can compute the correlation between $X_k$ and $X_{k'}$. But before we proceed with this calculation, we will discuss a little more about the covariance matrix.

We have that, if the assumption of conditional independence is satisfied, than the covariance between two coordinates, computed in (4.6), usually differs from the unconditional covariance computed in (4.5), except for some especial cases that will be illustrated in Example 4.1. Then, suppose for now that the unconditional general covariance matrix $\Sigma$, composed of the elements in (4.5), is equal to the covariance matrix $\Sigma_0$ given by (4.6) only when the conditional covariances are zero. We will present a technique to check if these two matrices are close, and if this is the case, we say it is reasonable to assume that the coordinates are at least conditionally uncorrelated.

Let $\mathbf{S}$ denote the usual sample covariance matrix, and $\mathbf{S}_0$ denote the covariance matrix computed in (4.6) with $\lambda$s and $\mu_{jl}$s being estimated by the npEM algorithm. In other words, $\mathbf{S}$ is an estimate for the general covariance matrix $\Sigma$, and $\mathbf{S}_0$ is an estimate for $\Sigma_0$, which is the covariance matrix under conditional independence assumption. To compare $\mathbf{S}$ and $\mathbf{S}_0$, we will use the average ratio of their eigenvalues, that is,

$$R = \frac{1}{r} \sum_{k=1}^{r} \frac{\delta_k}{\delta^0_k},$$  \hspace{1cm} (4.7)

where $\delta_k$ and $\delta^0_k$ denotes the $k$th eigenvalues of $\mathbf{S}$ and $\mathbf{S}_0$, respectively. If the two matrices are equal, we expect the eigenvalues to be the same, and then this average ratio should be 1.
Therefore, for each dataset, we will take bootstrap samples and construct an approximate 95% confidence interval for $R$. If the confidence interval contains 1, we cannot reject the assumption that the coordinates are conditionally uncorrelated. However, we have to keep investigating, since there are especial cases where the matrices $S$ and $S_0$ are the same but the coordinates are conditionally correlated.

Let us then go back to the point where we were computing the conditional covariances, and show how to compute the unconditional correlation between two coordinates.

1. Correlation between Conditionally Independent Coordinates

Under the conditionally independent assumption, the conditional correlation is zero, i.e $\text{Corr}(X_k, X_{k'}|j) = 0$, for $j = 1, 2$. In this case, the unconditional correlation is:

$$
\rho_{kk'}^{\text{ind}} = \frac{\text{Cov}(X_k, X_{k'})}{\sqrt{\text{Var}(X_k)\text{Var}(X_{k'})}} = \frac{\lambda(1-\lambda)(\mu_{1k} - \mu_{2k})(\mu_{1k'} - \mu_{2k'})}{\sqrt{\text{Var}(X_k)\text{Var}(X_{k'})}},
$$

(4.8)

where $\text{Var}(X_k)$ is given by (4.3).

2. Correlation between Conditionally Dependent Coordinates

Denote the conditional intra-subject correlation as $\rho_{kk'}^{(j)}$, which is

$$
\rho_{kk'}^{(j)} = \text{Corr}(X_k, X_{k'}|m) = \frac{\sigma_{kk'}^{(j)}}{\sqrt{\text{Var}(X_k)\text{Var}(X_{k'})}},
$$

(4.9)

for $m = 1, 2$. Hence, the unconditional correlation is

$$
\rho_{kk'} = \frac{\text{Cov}(X_k, X_{k'})}{\sqrt{\text{Var}(X_k)\text{Var}(X_{k'})}}
$$

$$
= \frac{\lambda \sigma_{kk'}^{(1)} + (1-\lambda)\sigma_{kk'}^{(2)} + \lambda(1-\lambda)(\mu_{1k} - \mu_{2k})(\mu_{1k'} - \mu_{2k'})}{\sqrt{\text{Var}(X_k)\text{Var}(X_{k'})}}
$$

$$
= \lambda \rho_{kk'}^{(1)} + (1-\lambda)\rho_{kk'}^{(2)} + \rho_{kk'}^{\text{ind}}
$$

$$
= \rho_{kk'}^{\text{ind}} + \rho_{kk'}^{\text{adj}}
$$

(4.10)
Elmore (2003) presented similar calculations for the binomial mixture. He also use the term “adj” in $\rho_{kk'}^{adj}$ to emphasize that the unconditional correlation is adjusted by this amount in the presence of intra-subject correlation. Then, the intuition here would be to test if $\rho_{kk'}^{adj}$ is zero to check if the coordinates are conditionally uncorrelated, which does not necessarily imply that they are conditionally independent. However, as we mentioned before, if the coordinates are conditionally correlated, then we know that the coordinates cannot be conditionally independent.

Before proceeding with the test, we need to be aware of an important point. Notice that

$$
\rho_{kk'}^{adj} = \lambda \rho_{kk'}^{(1)} + (1 - \lambda) \rho_{kk'}^{(2)}
$$

can be zero even when the conditional correlations are not, that is, if $\rho_{kk'}^{(j)} \neq 0$, for $m = 1, 2$, then

$$
\rho_{kk'}^{adj} = \lambda \rho_{kk'}^{(1)} + (1 - \lambda) \rho_{kk'}^{(2)} = 0 \quad \Rightarrow \quad \frac{\lambda}{1 - \lambda} = -\frac{\rho_{kk'}^{(2)}}{\rho_{kk'}^{(1)}},
$$

for all $k$ and $k'$ in $\{1, \ldots, r\}$, such that $k \neq k'$.

This shows that it is possible have two different mixtures, one where the coordinates are conditionally independent and one where the coordinates are not, that will have the same unconditional mean vectors and variance-covariance matrices, which also results in same unconditional correlation matrices. This does not mean that this model is not identifiable, because the mixtures density functions are not the same. We will see a simple example to illustrate what we just discussed.

**Example 4.1.** In this example we will compare the unconditional correlation matrices of two trivariate normal mixtures. The first mixture will have conditionally independent coordinates and the second one will present some correlation. Consider the model (4.1) with $\lambda = 0.5$, $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are both trivariate normal densities with means $\mathbf{\mu}_1^t = (0, 0, 0)$ and $\mathbf{\mu}_2^t = (5, 5, 5)$. In the first mixture, the covariance matrices for both components are simply the identity matrix, i.e $\Sigma_1^{ind} = \Sigma_2^{ind} = \mathbf{I}_3$. For the second
mixture, the coordinates are equi-correlated, with covariance matrices

\[
\Sigma_1 = \begin{pmatrix}
1 & 0.4 & 0.4 \\
0.4 & 1 & 0.4 \\
0.4 & 0.4 & 1
\end{pmatrix}
\quad \text{and} \quad
\Sigma_2 = \begin{pmatrix}
1 & -0.4 & -0.4 \\
-0.4 & 1 & -0.4 \\
-0.4 & -0.4 & 1
\end{pmatrix}.
\] (4.12)

In both mixtures, the coordinates are identically distributed. Then, using formulas (4.2), (4.3), and (4.5) the unconditional means and variances are:

\[
E(X_k) = 2.5, \quad Var(X_k) = 7.25 \quad \text{and} \quad Cov(X_k, X_{k'}) = 6.25, \quad (4.13)
\]

for \( k \) and \( k' \) in \( \{1, 2, 3\} \) and \( k \neq k' \). Hence, since \( \rho_{kk'}^{adj} = 0 \), we have

\[
\rho_{kk'} = \rho_{kk'}^{ind} = \frac{6.25}{7.25} = 0.862 \quad (4.14)
\]

Although the first two unconditional moments are the same, the two mixture densities are not, which means that the first two moments are not enough to describe the entire shape of a mixture of multivariate normals. Figures 4.1 and 4.2 allow us to visualize the difference between the two mixtures for the first two coordinates.

![Conditionally Independent](image1)

\[ \mu_1 = 0, \mu_2 = 5, \sigma^2 = 1, \rho_1 = 0, \rho_2 = 0 \]

![Conditionally Dependent](image2)

\[ \mu_1 = 0, \mu_2 = 5, \sigma^2 = 1, \rho_1 = 0.4, \rho_2 = -0.4 \]

Figure 4.1. Bivariate Normal Mixtures - Density Functions
Figure 4.1 presents the mixture densities, while Figure 4.2 shows the contour plots. In both figures, the plot on the left represents the conditionally independent case, and on the right, we have the conditionally dependent coordinates. It is clear that the two mixtures are not identical.

In this case, we need more than just the first two unconditional moments to distinguish the two cases. We decided then to check the third order moments, i.e. $E(X_k X_{k'}^2)$. For the normal mixture, we have

$$
E(X_k X_{k'}^2) = \lambda E_1(X_k X_{k'}^2) + (1 - \lambda) E_2(X_k X_{k'}^2) \\
= \lambda \left( \mu_{1k}^2 \sigma_{1k'}^2 + \mu_{1k'}^2 \right) + 2 \mu_{1k} \sigma_{k}^{(1)} \\
+ (1 - \lambda) \left( \mu_{2k}^2 \sigma_{2k'}^2 + \mu_{2k'}^2 \right) + 2 \mu_{2k} \sigma_{kk'}^{(1)} \tag{4.15}
$$

Applying formula (4.15) to our example, we have $E(X_k X_{k'}^2) = 65$ for the conditionally independent mixture, and $E(X_k X_{k'}^2) = 63$ for the conditionally dependent case. So, the unconditional third order cross product moments are not the same for the two mixtures.
This example motivates the weighted bootstrap hypothesis test we will develop in Section 4.3.

4.3 Weighted Bootstrap Hypothesis Test

Following the same idea as Elmore et al. (2004), we could test

\[ H_0 : \rho_{kk'}^{adj} = 0, \text{ for all pairs } k, k' \]

\[ H_A : \rho_{kk'}^{adj} \neq 0, \text{ for at least one pair } k, k'. \]

If we fail to reject \( H_0 \), then in practice it may be reasonable to assume that the coordinates are conditionally uncorrelated. This would be appropriate if \( \rho_{kk'}^{adj} \) was zero only when the coordinates are conditionally independent, which is not true as we have seen in Example 3.1. So, in order to test the assumption of interest, we will use not only the unconditional second order moments, but also the third and fourth order moments. In Subsection 4.3.4 we include goodness of fit tests to supplement the moment tests discussed in this section.

For the moment tests, the test statistics will come essentially from the following block matrix:

\[
A = \begin{pmatrix}
A_1 & A_2 \\
A_3^t & A_3
\end{pmatrix} = \begin{pmatrix}
E(XX^t) & E(X(X^2)^t) \\
E(X^2X^t) & E(X^2(X^2)^t)
\end{pmatrix},
\]

where \( X = (X_1, \ldots, X_r)^t \), and \( X^2 = (X_1^2, \ldots, X_r^2)^t \). The matrices \( A_1, A_2, \) and \( A_3 \) represent the matrix of unconditional second, third, and fourth order moments respectively.

Given a dataset, we can estimate matrix \( A \), say \( \hat{A} \), by the respective sample moments. Then we apply the npEM algorithm and estimate the matrix \( A \) using weighted moments. This will be the estimate of the unconditional moments under the null hypothesis and it will be called \( \hat{A}_0 \). If \( \hat{A}_0 \) approaches \( \hat{A} \), then the assumption of conditional
independence may be reasonable. Now we need to measure how “close” these two estimates are, and this is our test statistic, which is given by

\[
M = \sum_{1 \leq k,k' \leq r, k < k'} \frac{(\overline{X_kX_{k'}} - \hat{E}(X_kX_{k'}))^2}{\hat{E}(X_kX_{k'})} + \sum_{1 \leq k,k' \leq r, k \neq k'} \frac{(\overline{X_k^2X_{k'}^2} - \hat{E}(X_k^2X_{k'}^2))^2}{\hat{E}(X_k^2X_{k'}^2)} + \sum_{1 \leq k,k' \leq r, k < k'} \frac{(\overline{X_kX_{k'}} - \hat{E}(X_kX_{k'}))^2}{\hat{E}(X_kX_{k'})},
\]

where the overline represents a sample moment and and \(\hat{E}\) the expectation estimated by our model.

In order to assess the significance of the test statistic, \(M\), we suggest a weighted nonparametric bootstrap from the “best fitting” null model, \(\hat{g}_0(x)\), resulting from the npEM algorithm:

\[
\hat{g}_0(x_i) = \sum_{j=1}^m \lambda_j \prod_{k=1}^r \hat{f}_{jk}(x_{ik})
\]  

This bootstrap would be much simpler if \(\hat{f}_{jk}(\cdot)\) were a known density function, but in our case, \(\hat{f}_{jk}(\cdot)\) is a kernel density estimate. We need to know how to obtain a sample from a density estimate. Next we show how to do that in the non-mixture case first, and then we present how to obtain a sample from a mixture of kernel density estimates, which we call a weighted bootstrap, since the bootstrap involves the posterior probabilities.

### 4.3.1 Sampling from a Kernel Density Estimate

Given a sample \(X_1, \ldots, X_n\) from an unknown density \(f\), suppose that we want to generate a sequence of independent observations \(X_1^*, \ldots, X_n^*\) from \(f\). This would be impossible to achieve in practice because full information on \(f\) is not available. The usual approach to the problem is one of the two extremes:
• The simplest method is the naive resampling (bootstrap), where new observations are generated from the empirical distribution $F_n$ by successively selecting uniformly with replacement from the original sample $X_1, \ldots, X_n$. However, if the sample comes from a continuous distribution, this method has the obvious disadvantage that only a small number of different values can be generated.

• If we assume that $f$ comes from a parametric family indexed by a vector of parameters, then the original sample can be used to estimate the unknown parameters, and standard procedures can be used to generate random variables from a known distribution.

An intermediate approach to generate new observations from $f$ based on density estimation is presented in Silverman (1986), in which the sample $X_1, \ldots, X_n$ is used to construct a nonparametric density estimate $\hat{f}$ of the density $f$, and then as many independent realizations as required can be drawn from $\hat{f}$. If we use a kernel density estimate to obtain $\hat{f}$, the procedure described by Silverman (1986) is the same as the smoothed bootstrap from the statistics literature, which is a simple modification of the naive resampling method. This method does not only resample but adds some noise to the resampled observations. The noise is a continuous random variable with expectation 0 and small variance. Although we are generating observations from $\hat{f}$, we will see that it is not even necessary to find $\hat{f}$ explicitly in the simulation procedure.

In the univariate case, suppose that the sample $X_1, \ldots, X_n$ was used to obtain $\hat{f}$ by the ordinary kernel method with kernel function $K$, which is positive and integrates to one, and the bandwidth $h$ computed by the usual Silverman’s rule of thumb. In this thesis, we always use $K$ as the standard normal density.

In order to see the intuition behind the smoothed bootstrap, we need to rewrite the kernel density estimate $\hat{f}$, at some point $u$, as

$$
\hat{f}(u) = \frac{1}{nh} \sum_{i=1}^{n} K \left( \frac{u - X_i}{h} \right) \\
= \int \frac{1}{h} K \left( \frac{u - x}{h} \right) dF_n(x). \quad (4.19)
$$
Let $X_I$ and $V$ be independent random variables with $X_I \sim F_{n}(x)$ and $V \sim \frac{1}{h}K \left( \frac{u}{h} \right)$. Then $\hat{f}(u)$ is the pdf of $U = X_I + V$.

**Smoothed Bootstrap:** Algorithm to generate an observation from a kernel density estimate:

Step 1. Generate a random integer $I$ uniformly distributed on \{1, 2, \ldots, $n$\}.

Step 2. Generate a random variate $\varepsilon$ from the probability density function $K$ with scale parameter equals to 1.

Step 3. Return $X^* = X_I + h\varepsilon$.

This algorithm can be repeated as many times as necessary to give independent realizations $X^*_1, X^*_2, \ldots$ from $\hat{f}$. It guarantees that the density function of the new sample approximates the true density $f$ as well as possible with respect to the MISE.

Now we extend this algorithm to generate a sample from a nonparametric mixture density estimate.

### 4.3.2 Smoothed Weighted Bootstrap

Assume that we have a nonparametric multivariate mixture density and we want to generate observations from this mixture. Here, we extend the smoothed bootstrap algorithm described above to generate a sample from a nonparametric mixture density estimate. We call this algorithm *smoothed weighted bootstrap*.

Suppose that we have a sample $X_1, \ldots, X_n$, where each observation vector $X_i$ is a random vector with $r$ coordinates. Using the npEM algorithm we estimate the mixture:

$$
\hat{g}(x_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} \hat{f}_{jk}(x_{ik}),
$$

(4.20)

where $\hat{f}_{jk}(\cdot)$ is a weighted kernel density estimate. We also estimate the posterior probabilities, $\hat{p}_{ij}$, and the smoothing parameters, $\hat{h}_{jk}$, for $1 \leq i \leq n$, $1 \leq j \leq m$, and
\[ 1 \leq k \leq r, \text{ using the values obtained by equations 2.6 and 2.13, respectively, in the last iteration of the algorithm.} \]

**Smoothed Weighted Bootstrap:** Algorithm to generate an observation from a nonparametric mixture density estimate:

Step 1. Generate a random variable \( z^* = (z_1^*, \ldots, z_m^*) \) from a Multinomial(1, \( \text{prob}=(\hat{\lambda}_1, \ldots, \hat{\lambda}_m) \)), and suppose \( c \in \{1, \ldots, m\} \) to be such that \( z_c^* = 1 \).

Step 2. For each \( k \in \{1, \ldots, r\} \), repeat the next three steps:

1. Generate a random integer \( I_k \) distributed on \( \{1, 2, \ldots, n\} \) with probabilities \( (\hat{p}_1c, \hat{p}_2c, \ldots, \hat{p}_nc) \).
2. Generate a random variate \( \varepsilon_k \) from the probability density function \( K \) with scale parameter equals to 1.
3. Return \( X_k^* = X_{I_k} + h_{ck}\varepsilon_k \).

Step 3. Return \( X^* = (X_1^*, X_2^*, \ldots, X_r^*) \).

The algorithm above can be repeated as many times as necessary to obtain a new sample \( X_1^*, X_2^*, \ldots \).

### 4.3.3 Bootstrapped p-value

Using the smoothed weighted bootstrap described above, we can resample from the null model (4.18) to get a new sample \( X_1^*, \ldots, X_n^* \), and compute \( M^* \) using (4.17). We repeat this process a large number of times, say \( B \), and compute the bootstrapped p-value as

\[
\text{p-value} = \frac{\#(M^* \geq M)}{B},
\]

where \( M \) is the test statistic computed from the original sample.

### 4.3.4 Goodness of fit tests

The test we proposed in Section 4.3 compares high order empirical moments with moments estimated under the assumption of conditional independence. Besides that,
another possibility to check the assumption of conditionally independent coordinates would be to test the goodness of fit and that is what we will discuss in this section.

The basic idea is to compare the multivariate empirical cdf with the mixture cdf estimated by the npEM algorithm using some common criteria such as Kolmogorov-Smirnov or Cramér von Mises distances. The intuition behind this comparison comes from the fact that the empirical cdf only contains information given by the original dataset and the estimated mixture cdf is obtained under the assumption of conditional independence. If these two estimates are “close”, we have some evidence that the model fits and it may be reasonable to assume that the imposed assumption of conditional independence is correct.

Suppose that we have a sample $\mathbf{X}_1, \ldots, \mathbf{X}_n$, where each observation vector $\mathbf{X}_i$ is a random vector with $r$ coordinates. Let $\hat{F}_n(\cdot)$ be the multivariate empirical cdf as in equation (2.26). Using the npEM algorithm we estimate the mixture pdf by (4.20) and then the mixture cdf, say $\hat{G}(\cdot)$, is given by

$$
\hat{G}(\mathbf{x}) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} \hat{F}_{jk}(x_k),
$$

where $\hat{F}_{jk}(x_k) = \int_{-\infty}^{x_k} f_{jk}(x) dx$, that is, the estimated cdf for the $j$th component and $k$th coordinate at the point $x_k$.

Using the Kolmogorov-Smirnov or Cramér von Mises distances we compute the next two goodness of fit test statistics:

$$
KS = \sup_{1 \leq i \leq n} |\hat{F}_n(\mathbf{x}_i) - \hat{G}(\mathbf{x}_i)|
$$

$$
CvM = \int (\hat{F}_n(\mathbf{x}) - \hat{G}(\mathbf{x}))^2 d\hat{F}_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{F}_n(\mathbf{x}_i) - \hat{G}(\mathbf{x}_i))^2
$$

To find the p-value for each test, we follow the same procedure we used for the test to compare the moments, that is, we generate a bootstrap sample $\mathbf{X}_1^*, \ldots, \mathbf{X}_n^*$ using the smoothed weighted bootstrap algorithm and compute either $KS^*$ or $CvM^*$. We repeat
this process $B$ times and, depending on the test statistic we are using, we compute the bootstrapped p-value as

$$p\text{-value} = \frac{\#(KS^* \geq KS)}{B} \quad \text{or} \quad p\text{-value} = \frac{\#(CvM^* \geq CvM)}{B},$$

(4.25)

where $KS$ and $CvM$ are, respectively, the Kolmogorov-Smirnov and Cramér von Mises test statistics computed from the original sample.

Besides the two goodness of fit tests we presented above, we also investigated the behavior of several other statistics, such as a weighted version of the Cramér von Mises distance ($wCvM$), the integrated absolute difference ($D$), and one that resembles the Chi-square statistic ($C$). In other words,

$$wCvM = \frac{1}{n} \sum_{i=1}^{n} \frac{(\hat{F}_n(x_i) - \hat{G}(x_i))^2}{\max\{\hat{F}_n(x_i), 1 - \hat{F}_n(x_i)\}}$$

(4.26)

$$D = \int |\hat{F}_n(x) - \hat{G}(x)|d\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} |\hat{F}_n(x_i) - \hat{G}(x_i)|$$

(4.27)

$$C = \frac{n}{\hat{F}_n(x_i)} \sum_{i=1}^{n} (\hat{F}_n(x_i) - \hat{G}(x_i))^2$$

(4.28)

These tests gave us results similar to either the $KS$ or the $CvM$ test. Hence, we did not find it necessary to report results for all them. We will only use $KS$ and $CvM$ as our goodness of fit statistics.

In this chapter we presented several tools to check if the coordinates are at least conditionally uncorrelated. Let us summarize the techniques we discussed:

- Compare the empirical covariance matrix $S$ with the estimated covariance matrix $S_0$ under the assumption of conditional independence, using the average ratio of the respective eigenvalues computed in 4.7.

- The second tool is the $M$ test, which is based on the high order moments. The test statistic is computed in (4.17).
We also have the $K_S$ and $CvM$ tests, which are respectively based on the Kolmogorov Smirnov and Cramér von Mises distances. We compute the distances between the multivariate empirical cdf and the mixture cdf estimated by the npEM algorithm.

Now we will see the performance of each test for some simulated examples. Then we will apply the test to two real datasets.

### 4.4 Simulation with Known Null Model

We will report a small simulation to check the performance of the tests we proposed in the previous section in terms of their estimated size (when considering the usual significance level 0.05), and power.

In order to estimate the size of the test, we generate data from a mixture with conditionally independent coordinates, which we call the *null model*. Then we compute the test statistic and its respective p-value. We repeat this procedure say $S$ times, and find the proportion of p-values that fall below the significance level 0.05, that is, we estimate the proportion of tests that reject the null hypothesis when the null is true, and this is an estimate of the size of the test. To evaluate the power, we generate data from the *alternative model*, in which we set the coordinates to be correlated. In this case, the proportion of p-values less than 0.05 represents the proportion of times that the test rejects the null hypothesis when the null is false, which is the power of the test.

Consider the case where we are evaluating the power of the test. The next steps describe exactly what we are doing in this simulation.

Step 1. Generate $n$ observation vectors from the two component location normal mixture

$$g(x) = \lambda MN(x; \mu_1, I_r) + (1 - \lambda) MN(x; \mu_2, \Sigma_2), \quad (4.29)$$

with $\lambda = 0.4, \mu_1 = (0, \ldots, 0)^t, \mu_2 = (3, \ldots, 3)^t$, for $r = 3, 5, 8$ repeated measures. The covariance matrix for the first component is simply the identity matrix, and since this is the alternative model, we set the coordinates to be
equi-correlated in the second component

\[
\Sigma_2 = \begin{pmatrix}
1 & \rho & \ldots & \rho \\
\rho & 1 & \ldots & \rho \\
\vdots & \vdots & \ddots & \vdots \\
\rho & \rho & \ldots & 1
\end{pmatrix},
\]  \hspace{1cm} (4.30)

with \( \rho = 0.7 \).

Step 2. Compute the observed test statistics \( M, KS, \) and \( CvM \).

Step 3. Generate a new sample \( X_1^*, \ldots, X_n^* \) from the null model, which is given by equation 4.29 with \( \rho = 0 \).

Step 4. Using \( X_1^*, \ldots, X_n^* \), compute the test statistics \( M^*, KS^*, \) and \( CvM^* \).

Step 5. Repeat steps 3 and 4 for \( B \) times and find the p-value for each test, as shown in 4.21 and 4.25.

Step 6. Now we repeat steps 1-5 for \( S \) times, and for the usual significance level of 0.05, the power is estimated by

\[
\text{power} = \frac{\#\{\text{p-value} \leq 0.05\}}{S}
\]

We follow the same steps above to estimate the size of the test, with the only difference that the model 4.29 in Step 1 \( \Sigma_2 = I_r \).

In our simulation we took \( S = B = 100 \), and in Table 4.1 we have the estimated size and the power of the \( M, KS, \) and \( CvM \) tests for different numbers of repeated measures and sample sizes.

Notice that the estimated size was close to the size \( \alpha = 0.05 \) in most of the cases, and the observed power is at least 0.99. In Figure 4.3 we have the histograms for the p-values obtained for each test when the data was generated from the null model. In this case, we expect these histograms to look uniformly distributed, which in fact they do.
Figure 4.3. Histogram of the p-values under the null model when using simulation. Each row of histograms represents a different test, and each column is for a number of repeated measure (r = 3, 5, and 8). Histograms for M, KS, and CVM tests are in first, second, and third row, respectively.
Simulating data from the null model is only possible when we know the component densities, which is not true in practice. Also our method does not assume a parametric form for these densities. Hence, in order to produce a new sample from the null model in Step 3, we will use the smoothed weighted bootstrap algorithm described in 4.3.2. See Section 4.5 for more details and results.

### 4.5 Simulation Using the Weighted Smoothed Bootstrap

In this section, we will repeat the simulation from Section 4.4, but instead of generating data directly from the null model as described in Step 3, we will use the smoothed weighted bootstrap to obtain a new sample. Therefore, all the steps remain the same, except for Step 3, which is replaced by the new Step 3’:

Step 3’. Generate a new sample $X_1^*, \ldots, X_n^*$ from the null model using the smoothed weighted bootstrap algorithm given in 4.3.2.

Table 4.2 presents the estimates size and the power of the $M$, $KS$, and $CvM$ tests for different number of repeated measures and sample sizes, and Figure 4.4 shows the histograms for the p-values obtained when the data is generated from the null model.

<table>
<thead>
<tr>
<th>$r$</th>
<th>$n$</th>
<th><strong>Test M</strong></th>
<th><strong>Test KS</strong></th>
<th><strong>Test CvM</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>size</td>
<td>power</td>
<td>size</td>
</tr>
<tr>
<td>3</td>
<td>300</td>
<td>0.06</td>
<td>1.00</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>500</td>
<td>0.02</td>
<td>1.00</td>
<td>0.06</td>
</tr>
<tr>
<td>8</td>
<td>800</td>
<td>0.00</td>
<td>1.00</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 4.2. Estimated size and power of the tests from smoothed weighted bootstrap.
Figure 4.4. Histogram of the p-values under the null model when using the smoothed weighted bootstrap. Each row of histograms represents a different test, and each column is for a number of repeated measure (r = 3, 5, and 8). Histograms for $M$, $KS$, and $CvM$ tests are in first, second, and third row, respectively.
The first row of histograms in Figure 4.4 are associated with the $M$ test for $r = 3, 5,$ and $8$. We notice that, for the $M$ test, the histograms for both $r = 3$ and $5$ look roughly uniform. However, the histogram is skewed when we have $r = 8$ repeated measures. We observe the same thing for the $KS$ test (second row of histograms). In fact, for all three tests the distribution of the p-values gets skewed, as we increase the number of repeated measures, which implies that the size of the test decreases. For example, when $r = 8$, we have size 0 and power 1 for basically all tests, which means that we failed to reject the null hypothesis when it was true, and we rejected the null when it was false in every replication of our simulation.

We expected the histograms in this section to be similar to the histograms shown in Section 4.4, but they are not, especially when the number of repeated measures gets larger. The difference can only be associated with the smoothed weighted bootstrap. But the tests can still be used, since the size is very small and the power is one.

4.6 Comparing $\Sigma$ and $\Sigma_0$ using their eigenvalues

In this section we will show how the technique we proposed to compare the covariance matrices $\Sigma$ and $\Sigma_0$ performs. We will generate data from the same models we used in 4.4, and for each dataset, we will take $B = 100$ nonparametric bootstrap samples and compute the average ration of the eigenvalues for $S$ and $S_0$ as in equation (4.7). Hence, we compute a bootstrap 95% confidence interval (CI) for the average ratio of the eigenvalues. If the data is generated from a model with conditionally independent coordinates, we expect the confidence intervals to contain 1, whereas if the data is generated from a model with conditionally correlated coordinates, we expect the confidence interval not to contain 1.

Figure 4.5 shows the plots for 100 CIs computed for both the model with conditionally independent coordinates (plot on the left) and the model with conditionally correlated coordinates (plot on the right). The confidence intervals in red represent those that did not contain 1, when they should. However, because we are computing 95% CIs, we expect about 5% of the CIs do not cover 1 on the left plot, and that is what we observe.
Figure 4.5. Bootstrap 95% CIs for the average ratio of the eigenvalues for $S$ and $S_0$. On the left we have the CIs when the data has conditionally independent coordinates, and on the right we have the conditionally non-independent case. The dashed line represents our target value 1. Each dataset have $n = 300$ observations and $r = 3$ repeated measures.

4.7 Checking the Conditional Independence Assumption in Real Data

We will apply the tools discussed in this chapter to check if the assumption of conditional independence may be reasonable for the two real datasets we are analysing in this thesis: the Water-level task data introduced in 2.6.3, and the RT data presented in 2.6.4.

4.7.1 Waterdata

The waterdata contains $r = 8$ measurements for each subject, and assuming that the coordinates are not identically distributed, i.e $b_k = k$, $k = 1, \ldots, 8$. First of all, we compared the covariance matrices using the average ratio of the eigenvalues computed by (4.7). We observed $R = 0.922$ and the bootstrap 95% confidence interval is $(0.883, 0.948)$, which indicates that the coordinates are conditionally correlated. Figure 4.6 shows the scatterplot for the eigenvalues of $S$ and $S_0$. 
Figure 4.6. Eigenvalues of $S$ and $S_0$ computed for the Waterdata.

We also implemented the $M$ and $KS$ tests for which we generated $B = 1000$ smoothed weighted bootstrap samples and computed the respective p-values. The p-value is 0.007 for the $M$ test, and 0.012 for the $KS$ test. In both tests, considering the 0.05 significance level, we reject the null hypothesis. This means that the conditional independence assumption may not be reasonable for this dataset. Figure 4.7 shows the sampling distribution of test statistic for each test, and the dashed lines represents the observed test statistic computed from the Waterdata.

Figure 4.7. Sampling distribution of $M$ and $KS$ test statistics. The dashed line represents the observed value of $M$ and $KS$ computed from the Waterdata.
4.7.2  RTdata

The RTdata contains $r = 6$ measurements for each subject, and assuming that the coordinates are not identically distributed, we compared the covariance matrices using the average ratio of the eigenvalues. We observed $R = 0.962$ and the bootstrap 95% confidence interval is $(0.924, 0.981)$. Although the confidence interval does not contain 1, it is very close to 1. This suggests further tests, but it is possible that the coordinates are conditionally uncorrelated. Figure 4.8 shows the scatterplot for the eigenvalues of $S$ and $S_0$. Notice that most of the eigenvalues lie on the identity line, except for the largest one. Then, it could be the case where, if we exclude the coordinate related to the largest eigenvalue in the covariance matrices, the remaining would be conditionally uncorrelated.

![Figure 4.8. Eigenvalues of $S$ and $S_0$ computed for the RTdata.](image)

For the $M$ and $KS$ tests, we generated $B = 1000$ smoothed weighted bootstrap samples and computed the respective p-values. The p-value is 0.824 for the $M$ test, and 0.046 for the $KS$ test. Thus, considering the 0.05 significance level, we fail to reject the null hypothesis in the $M$ test, but we reject it for the $KS$ test. Figure 4.9 shows the sampling distribution of test statistic for each test, and the dashed lines represents the observed test statistic computed from the Waterdata.
Since we are using these tests as tools to check the assumption in our model, and basically two of them failed to reject the hypothesis of unconditional correlation, it may be reasonable to assume that the coordinates in the RT data are conditionally independent.

Figure 4.9. Sampling distribution of $M$ and $KS$ test statistics. The dashed line represents the observed value of $M$ and $KS$ computed from the RT data.
Chapter 5

Data Analysis

5.1 Introduction

In this chapter we will apply the npEM algorithm proposed in Chapter 2 to two real datasets, the water-level task data (Waterdata) and the RT task data (RTdata), introduced in subsections 2.6.3 and 2.6.4, respectively.

We will explore the full capability of our algorithm, fitting the data sets with the most general model, where the coordinates are conditionally independent but not identically distributed, and also show how the blocking structure makes sense in these real situations. We will see the estimated component pdfs and cdfs for each model, and although our algorithm does not estimate parameters, we will compute the component means and variances from the final component density functions and provide these estimates as additional information about each subpopulation. When possible, we will interpret the components in terms of the real problem.

5.2 Waterdata

The experiment was described in Section 2.6.3, and for more details see Thomas et al. (1993). In summary, we have 405 children aged 11 to 16 years that were subjected to the water-level task. Each child was presented with a simple sketch of a rectangular vessel with a cap, each tilted to one of the eight clock-hour orientations: 11, 4, 2, 7, 10, 5, 1, and 8 o’clock (in order that appears in the dataset). The response variable is the acute angle, measured in degrees, between the line drawn by the child and the horizontal line (correct orientation) defined as zero degrees, with sign according to the slope of the line.
In 4.7.1 we applied the tools to check unconditional correlation for this dataset, and although all the results indicate that the coordinates are conditionally correlated, we will still proceed with the analysis using the npEM algorithm.

Elmore (2003) analysed this dataset using the cutpoint model for three and four components. If we consider the four component case and we apply the npEM algorithm assuming that the coordinates are conditionally iid, we get very similar results (see Figure 5.1). The interpretation given in Elmore (2003) for the four groups of respondents with respect to their ability to solve the water-level task can also be used here. There is a group representing 11.3% of the population (green line on the plot) that seems to be placing the horizontal line at angles uniformly throughout the range of possible values. There is another group representing 38.3% of the population (red line) who grasp the concept of the task, but there is some variation about the horizontal line. The group composed by 45.4% of the population (blue line) represents those who actually know the concept of the task and draw a line at or very near the horizontal line every time. The last group represents only 5% of the population and is composed of those who place the line perpendicular to the vessel bottom, that is, they draw the line at approximately -60°, -30°, 30°, and 60°.

![Figure 5.1](image-url)  
*Waterdata component pdfs and cdfs for a four component mixture assuming conditionally iid coordinates.*
Notice that the four component mixture with conditionally iid coordinates is basically a pure scale mixture. Although we have a nice interpretation for each group in this mixture, this model is too restrictive. We will explore the full capability of our model and fit a mixture using the most general model given by 2.2, that is, the coordinates are independent but not identically distributed. In Section 2.6.3 we applied the minimum distance method, described in Section 2.6, to select the number of components for this dataset, and we saw that three components \((m = 3)\) resulted in a better fit. Figure 5.2 shows the components cdfs for each coordinate.

![Figure 5.2](image)

Figure 5.2. Waterdata component cdfs for a three component mixture with conditionally independent coordinates.

The first observation in Figure 5.2 is that the coordinates, conditional on the component membership, are not all identically distributed. In fact, this statement is about two of the components (red and green lines), since the component represented by the blue line, which is basically a degenerate distribution around zero, seems the same for every coordinate. This component represents 47.6% of the population, and it is composed of those who know the concept of the task. A second observation is that component cdfs for some pairs of the coordinates are very similar, and this suggests that we should group
them into blocks of conditionally iid coordinates. Notice that the pairs of coordinates (2,5) and (3,8) are basically identical. There are also similarities between the pairs of coordinates (4,7) and (1,6). The interesting thing about this pairing structure is that the pairs (4,7), (3,8), (2,5), and (1,6) correspond respectively to (1 and 7 o’clock), (2 and 8 o’clock), (4 and 10 o’clock), and (5 and 11 o’clock).

Based on this, we fitted a new model using the npEM algorithm with blocks of conditionally independent coordinates, as described in the most general model (2.4). In Figure 5.3 we present the estimated component cdfs, and in addition, we show the estimated component pdfs in Figure 5.4.

![Waterdata component cdfs](image)

Figure 5.3. Waterdata component cdfs for a three component mixture with four conditionally independent blocks of coordinates.
What we see now in Figure 5.3 is that component distributions among blocks are all different. However, the component represented by the blue line, which represents 48% of the population in this setting, remains intact among blocks. Observing the component pdfs in Figure 5.4, in terms of the blocks of conditionally iid coordinates, we notice that $f_{j1}(x) = f_{j4}(-x)$, and $f_{j2}(x) = f_{j3}(-x)$, for $j = 1, 2, 3$. In other words, the top left plot looks like the mirror of the bottom right plot with respect to zero. Relating this to the clock orientation, coordinates 1 and 6 represent 11 and 5 o’clock, respectively, whereas coordinates 4 and 7 represent 7 and 1 o’clock, respectively, which are exactly the mirror clock orientations with respect to the horizontal line. The same thing happens with the top right and the bottom left plots, which represent (4 and 10 o’clock) and (2 and 8 o’clock), respectively.

We include in Table 5.1 the estimates of the component means and variances, with their respective bootstrap standard errors. Notice that we have two bootstrap standard errors. The first one is the usual nonparametric bootstrap standard errors, and the second one is a weighted bootstrap, where a new sample is obtained based on the model estimated by the npEM algorithm. For the component means, Elmore (2003) provided, in Section 3.4.4 of his thesis, a formula to compute the estimated standard errors. He provides the formula for the conditionally iid case, and we adapted it for the case with blocks of conditionally iid coordinates, which yields the formula

$$\tilde{SE}(\hat{\mu}_{j\ell}) = \sqrt{\frac{\hat{\sigma}^2_{j\ell}}{nC_\ell\hat{\lambda}_j}},$$  \hspace{1cm} (5.1)$$

for $1 \leq j \leq m$, $1 \leq \ell \leq B$, where $C_\ell$ denotes the number of coordinates in the $\ell$th block, and $n\hat{\lambda}_j$ is the estimated sample size for the $j$th component. Note that this formula resembles the standard error for the sample mean. Elmore (2003) provides evidence that this estimate works fairly well though generally biased downward.

We included the estimated standard errors for the component means using equation 5.1 in Table 5.1 as well, and we see that they are close to the weighted bootstrap standard errors. Table 5.1 reinforces the results shown in Figures 5.3 and 5.4 that $f_{j1}(x) = f_{j4}(-x)$, and $f_{j2}(x) = f_{j3}(-x)$, for $j = 1, 2, 3$. The estimated component
means for block one are basically the negative of the component means for block four, and the component variances are relatively close. The same comparison applies to blocks two and three.

5.3 RTdata

This dataset was introduced in Section 2.6.4, and it is described in details in Miller et al. (2001) and Cruz-Medina et al. (2004). In summary, we have the reaction time (RT), in milliseconds, of 197 children who provided correct responses for $r = 6$ task
trials, where in each trial, the child is presented with two visual stimuli displayed on a
computer monitor, and he/she must press one key if the right stimulus matches the left
or another key if the right stimulus is a mirror image.

In 4.7.2 we applied the tools to check unconditional correlation, and basically two
of tests, the comparison between covariance matrices and the $M$ test, failed to reject the
hypothesis of unconditional correlation. Therefore, it seems reasonable to assume that
the coordinates composing this dataset are conditionally independent.

We will start analyzing this dataset using the most general model given by (2.2),
where each coordinate can assume a different density function $f_{jk}$ for $k = 1, \ldots, 6$. In
Section 2.6.4 we applied the minimum distance method from Section 2.6 to this dataset,
and three ($m = 3$) is the number of components that best fits the data. In Figure 5.5 we
have the estimated component pdfs for each coordinate, and the plots look very similar,
which may suggests that we have conditionally iid coordinates.
Figure 5.5. RTdata component cdfs for a three component mixture with conditionally independent coordinates.

Table 5.2 presents the estimated component means and standard deviations for each coordinate. The first component represents only approximately 5% of the population, and is composed of those who take longer to press the right key (3.5 to 4 seconds). The second component represents those who take about 2.3 seconds to press the right key, and 44% of the population belongs to this subgroup. The majority (51% of the population) are in the third component, which are composed of those who press the right key faster than the others (1.5 seconds). The variability for this group is considerably smaller than the other two.

Now we will proceed with the analysis assuming that the coordinates are conditionally iid. In Figure 5.6 we show the component pdfs and cdfs, and Table 5.3 contains the component means and standard deviations. Notice that the results here are not very different than the previous analysis, which reinforces that the coordinates in this dataset might be conditionally iid.

The interpretation for this mixture is basically the same as the one we gave before for the model with conditionally independent and non identically distributed coordinates,
with the difference that components one, two, and three now represent 17%, 49.5%, and 34.5% of the population, respectively. The first group takes longer to press the right key (about 3 seconds), and it has more variability than the other two. Children in the second component press the key in about 2 seconds. The third group is composed of those who press the right key in 1.5 seconds on average, with very small standard deviation.

Table 5.2. RTdata: estimates and standard errors for the component means and std. deviations for a three component mixture with conditionally independent coordinates.

<table>
<thead>
<tr>
<th>Component Means</th>
<th>Component Std. Deviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Par</td>
<td>Estimate</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>3488.563</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>3579.512</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>3739.768</td>
</tr>
<tr>
<td>$\mu_{14}$</td>
<td>4127.264</td>
</tr>
<tr>
<td>$\mu_{15}$</td>
<td>4046.005</td>
</tr>
<tr>
<td>$\mu_{16}$</td>
<td>3701.862</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>2679.30</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>2349.817</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>2390.346</td>
</tr>
<tr>
<td>$\mu_{24}$</td>
<td>2309.429</td>
</tr>
<tr>
<td>$\mu_{25}$</td>
<td>2301.434</td>
</tr>
<tr>
<td>$\mu_{26}$</td>
<td>2277.970</td>
</tr>
<tr>
<td>$\mu_{31}$</td>
<td>1746.909</td>
</tr>
<tr>
<td>$\mu_{32}$</td>
<td>1527.054</td>
</tr>
<tr>
<td>$\mu_{33}$</td>
<td>1538.097</td>
</tr>
<tr>
<td>$\mu_{34}$</td>
<td>1534.455</td>
</tr>
<tr>
<td>$\mu_{35}$</td>
<td>1535.099</td>
</tr>
<tr>
<td>$\mu_{36}$</td>
<td>1647.961</td>
</tr>
</tbody>
</table>

Table 5.3. RTdata: estimates and standard errors for the component means and standard deviations for a three component mixture with conditionally independent coordinates.
Figure 5.6. RTdata component pdfs and cdfs for a three component mixture with conditionally iid coordinates.
Chapter 6

Discussion and Future Work

6.1 Introduction

The main goal of this thesis is to provide a complete methodology to analyze finite multivariate mixture models. Because our approach is fully nonparametric, it is fundamental to assume that the coordinates are independent conditional on the component membership in order to have an identifiable mixture. Therefore, we presented an EM-like algorithm to fit a nonparametric mixture and some tools that can be useful to check if the main assumption is met.

In this last chapter we will summarize the main points presented in this thesis and point out some of the questions that arose throughout the course of this research that can determine the directions of this work in the future.

6.2 Discussion

In Chapter 1 we introduced the finite multivariate mixture problem and, although identifiability is only partially settled for our model, we discussed some of the main results existing in the literature.

Chapter 2 contains the main part of this thesis, that is the nonparametric EM (npEM) algorithm, which is an extension of the semiparametric stochastic EM of Bordes et al. (2007) to the multivariate case. This algorithm is a improved version of the algorithm presented Benaglia et al. (2008), since we allow a distinct bandwidth for each coordinate and component in the nonparametric density estimation step. The npEM procedure results in estimates of the mixing proportions and the component density functions in the mixture, from which we can estimate not only the component cdfs but also any parameters within a component, such as component means and standard deviations. Because our model assumes that the number of components is known, we present a
method to select the number of components. In Section 2.6 we adapted the minimum distance method introduced in Chen and Kalbfleisch (1996) to the nonparametric mixture case, which is essentially a penalized Kolmogorov Smirnov distance. Simulations showed that this method was accurate in terms of selecting the right number of components.

In Chapter 3 we evaluated the performance of the npEM algorithm using Monte Carlo simulations. We simulated from pure location, pure scale, and asymmetric mixtures, and we had models with conditionally iid coordinates, models with blocks of conditionally iid coordinates, and also models where the coordinates come from very distinct density functions. Some models were chosen specifically from examples used by other authors, so we could compare our method to those existing in the literature. When the data was simulated from a mixture of normal distributions, the npEM performed as well as the normal EM, which is a reasonable gold standard for comparison, since it is the parametric procedure that assumes the correct model. In some cases such as mixtures of Laplace or asymmetric distributions, the npEM produced much better results than the normal EM. In comparison to the cut-point model (Elmore, 2003), and the exponential tilt EM (Qin and Leung, 2006; Wrobél, 2008), the npEM presented similar results. However, we insist on emphasizing that there is still an advantage for our algorithm, since we provide a fully nonparametric approach and only assume conditional independence for the components. In the last simulation in Chapter 3, we compared the npEM algorithm with the inversion method of Hall et al. (2005), and our procedure dramatically outperforms the inversion method.

Our algorithm is computationally simpler than all the algorithms presented in the literature. Another advantage is that none of the methods existing in the literature to estimate multivariate mixtures with conditionally iid coordinates are able to estimate this model assuming the blocking structure. They would either assume that all the coordinates are conditionally iid or they all have different distributions.

The nonparametric multivariate mixture is dependent upon the assumption that the coordinates within an observation vector are independent conditional on the component membership. Therefore, in Chapter 4, we presented some tools to check this assumption. Notice that we do not actually test if the data is conditionally independent.
We only provide tools that can be used to check if it is reasonable to assume that the coordinates are at least conditionally uncorrelated.

In our first tool, we compare the empirical covariance matrix $S$ with the estimated covariance matrix $S_0$ under the assumption of conditional independence, using the average ratio of the respective eigenvalues computed in (4.7). The second method is the $M$ test, which is based on the comparison of higher order moments, and the p-value is obtained by bootstrapping. We also presented two more bootstrap-based hypothesis tests, the $KS$ and $CvM$ tests, which are respectively the Kolmogorov Smirnov and Cramér von Mises distances. Simulations suggest that these are fairly powerful tools capable of detecting conditional correlation in the data.

In Chapter 5 we applied the npEM to two real datasets, the water-level task data and the RT data, introduced in subsections 2.6.3 and 2.6.4, respectively. The water-level task data was the motivation for some of the models we proposed in this thesis, especially the nonparametric multivariate mixture with blocks of conditionally iid coordinates. We provided a complete analysis of these two datasets with some interesting interpretations about the component distribution functions.

6.3 Future Work

The npEM algorithm with the blocking structure described in 2.3.1 is implemented in the R package mixtools. However, when we were analyzing the water-level task data in Section 5.2, we noticed that for one of the components, all the coordinates could be assumed conditionally iid, whereas the other two components would have blocks of conditionally iid coordinates. The model that allows this structure is already presented in this thesis (see 2.5.3), but it is not currently implemented in mixtools. We plan to include this model in mixtools shortly.

The other problem that arose when we were analysing the real datasets is how to choose the blocks of conditionally iid coordinates, and also how to check that the coordinates within a component are in fact identically distributed. For now, we use either the estimated component pdfs or cdfs to groups coordinates into groups that look similar, but that is not the best strategy. There are several procedures in the literature
to test if two vectors of data come from the same distribution. One idea would be to adapt one of these methods for the mixture case, so we can test within a component to see if coordinates may have the same distribution.
Appendix A

Extra Simulations Results from Chapter 4

A.1 Multivariate Normal Location Mixture in 3.2.4

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM Mean</th>
<th>Std.Error</th>
<th>Normal Mean</th>
<th>Std.Error</th>
<th>Exponential Tilt Mean</th>
<th>Std.Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.5</td>
<td>0.50181</td>
<td>0.03356</td>
<td>0.50180</td>
<td>0.03237</td>
<td>0.4988</td>
<td>0.0338</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0</td>
<td>0.00190</td>
<td>0.07526</td>
<td>0.00548</td>
<td>0.07392</td>
<td>-0.0077</td>
<td>0.0753</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>0</td>
<td>-0.00187</td>
<td>0.07400</td>
<td>0.00289</td>
<td>0.07357</td>
<td>-0.0022</td>
<td>0.0795</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0</td>
<td>0.04687</td>
<td>0.08265</td>
<td>0.04440</td>
<td>0.07924</td>
<td>0.0151</td>
<td>0.0971</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>1</td>
<td>1.00499</td>
<td>0.07456</td>
<td>1.00172</td>
<td>0.07593</td>
<td>1.0064</td>
<td>0.0718</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>1.5</td>
<td>1.50968</td>
<td>0.07568</td>
<td>1.50483</td>
<td>0.07409</td>
<td>1.5066</td>
<td>0.0772</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>2.5</td>
<td>2.46242</td>
<td>0.08938</td>
<td>2.50511</td>
<td>0.08620</td>
<td>2.4871</td>
<td>0.0895</td>
</tr>
<tr>
<td>$\sigma_{11}^2$</td>
<td>1</td>
<td>0.99606</td>
<td>0.09690</td>
<td>0.99885</td>
<td>0.09757</td>
<td>0.9950</td>
<td>0.0972</td>
</tr>
<tr>
<td>$\sigma_{12}^2$</td>
<td>1</td>
<td>0.97808</td>
<td>0.10050</td>
<td>0.98607</td>
<td>0.10043</td>
<td>0.9830</td>
<td>0.1158</td>
</tr>
<tr>
<td>$\sigma_{13}^2$</td>
<td>1</td>
<td>1.08968</td>
<td>0.14574</td>
<td>0.98843</td>
<td>0.12080</td>
<td>1.0298</td>
<td>0.1536</td>
</tr>
<tr>
<td>$\sigma_{21}^2$</td>
<td>1</td>
<td>0.99272</td>
<td>0.09964</td>
<td>0.99590</td>
<td>0.09711</td>
<td>0.9882</td>
<td>0.0955</td>
</tr>
<tr>
<td>$\sigma_{22}^2$</td>
<td>1</td>
<td>0.98944</td>
<td>0.10121</td>
<td>0.99494</td>
<td>0.10124</td>
<td>0.9866</td>
<td>0.1118</td>
</tr>
<tr>
<td>$\sigma_{23}^2$</td>
<td>1</td>
<td>1.09939</td>
<td>0.14617</td>
<td>0.99314</td>
<td>0.11390</td>
<td>1.0233</td>
<td>0.1631</td>
</tr>
</tbody>
</table>

Table A.1. Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t, \mu_2 = (1, 1, 5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.5$. The results are based on 500 simulations with sample size $n = 500$ each.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM</th>
<th>Normal</th>
<th>Exponential Tilt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.Error</td>
<td>Mean</td>
<td>Std.Error</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.8</td>
<td>0.80108</td>
<td>0.02738</td>
<td>0.80012</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0</td>
<td>-0.00399</td>
<td>0.05576</td>
<td>-0.00272</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>0</td>
<td>0.00336</td>
<td>0.05709</td>
<td>0.00490</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0</td>
<td>0.02804</td>
<td>0.06261</td>
<td>0.00142</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>1</td>
<td>1.01013</td>
<td>0.12360</td>
<td>1.00012</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>1.5</td>
<td>1.51603</td>
<td>0.13625</td>
<td>1.50263</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>2.5</td>
<td>2.40381</td>
<td>0.16837</td>
<td>2.50064</td>
</tr>
<tr>
<td>$\sigma_{11}^2$</td>
<td>1</td>
<td>0.99128</td>
<td>0.07513</td>
<td>0.99768</td>
</tr>
<tr>
<td>$\sigma_{12}^2$</td>
<td>1</td>
<td>0.99229</td>
<td>0.07712</td>
<td>1.00057</td>
</tr>
<tr>
<td>$\sigma_{13}^2$</td>
<td>1</td>
<td>1.05856</td>
<td>0.10129</td>
<td>1.00296</td>
</tr>
<tr>
<td>$\sigma_{21}^2$</td>
<td>1</td>
<td>0.99673</td>
<td>0.17881</td>
<td>0.98392</td>
</tr>
<tr>
<td>$\sigma_{22}^2$</td>
<td>1</td>
<td>0.99498</td>
<td>0.18055</td>
<td>0.98767</td>
</tr>
<tr>
<td>$\sigma_{23}^2$</td>
<td>1</td>
<td>1.25086</td>
<td>0.32600</td>
<td>0.98456</td>
</tr>
</tbody>
</table>

Table A.2. Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t$, $\mu_2 = (1, 1.5, 2.5)^t$, and $\sigma_1^2 = \sigma_2^2 = (1, 1, 1)^t$ when $\lambda = 0.8$. The results are based on 500 simulations with sample size $n = 500$ each.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM</th>
<th>Normal</th>
<th>Exponential Tilt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.Error</td>
<td>Mean</td>
<td>Std.Error</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.5</td>
<td>0.50450</td>
<td>0.02527</td>
<td>0.50049</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0</td>
<td>0.01062</td>
<td>0.06990</td>
<td>0.00390</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>0</td>
<td>0.00916</td>
<td>0.06617</td>
<td>0.00192</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0</td>
<td>0.02766</td>
<td>0.06687</td>
<td>0.00225</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>2</td>
<td>2.01008</td>
<td>0.08557</td>
<td>2.00087</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>2.5</td>
<td>2.51558</td>
<td>0.09472</td>
<td>2.50263</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>3</td>
<td>2.99904</td>
<td>0.32600</td>
<td>3.00083</td>
</tr>
<tr>
<td>$\sigma_{11}^2$</td>
<td>1</td>
<td>1.01025</td>
<td>0.09849</td>
<td>0.99748</td>
</tr>
<tr>
<td>$\sigma_{12}^2$</td>
<td>1</td>
<td>1.00557</td>
<td>0.09980</td>
<td>0.98774</td>
</tr>
<tr>
<td>$\sigma_{13}^2$</td>
<td>1</td>
<td>1.06009</td>
<td>0.12736</td>
<td>0.98852</td>
</tr>
<tr>
<td>$\sigma_{21}^2$</td>
<td>1.5</td>
<td>1.48232</td>
<td>0.13935</td>
<td>1.49613</td>
</tr>
<tr>
<td>$\sigma_{22}^2$</td>
<td>2</td>
<td>1.96696</td>
<td>0.18721</td>
<td>1.99122</td>
</tr>
<tr>
<td>$\sigma_{23}^2$</td>
<td>1</td>
<td>1.00707</td>
<td>0.10328</td>
<td>0.99840</td>
</tr>
</tbody>
</table>

Table A.3. Two-component multivariate normal mixture with $\mu_1 = (0, 0, 0)^t$, $\mu_2 = (2, 2.5, 3)^t$, $\sigma_1^2 = (1, 1, 1)^t$, and $\sigma_2^2 = (1.5, 2, 1)$ when $\lambda = 0.5$. The results are based on 500 simulations with sample size $n = 500$ each.
A.2 Mixture of Asymmetric Distributions in 3.2.5
Table A.5. Two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2),$ and $\beta_2 = (2, 1, 0.5)$ for $\lambda = 0.3$. The means and variances are $\alpha \beta$ and $\alpha \beta^2$, respectively. The results are based on 1000 simulations with sample size $n = 300$ each.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM Mean</th>
<th>Std.Error</th>
<th>Bias</th>
<th>Normal Mean</th>
<th>Std.Error</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.3</td>
<td>0.2849</td>
<td>0.0313</td>
<td>-0.0151</td>
<td>0.2465</td>
<td>0.0402</td>
<td>-0.0535</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>4</td>
<td>3.8953</td>
<td>0.3371</td>
<td>-0.1047</td>
<td>3.5739</td>
<td>0.3900</td>
<td>-0.4261</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>4</td>
<td>3.8742</td>
<td>0.3652</td>
<td>-0.1258</td>
<td>3.4763</td>
<td>0.4514</td>
<td>-0.5237</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>4</td>
<td>3.8649</td>
<td>0.3438</td>
<td>-0.1351</td>
<td>3.5485</td>
<td>0.6193</td>
<td>-0.4515</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>10</td>
<td>9.8961</td>
<td>0.3234</td>
<td>-0.1039</td>
<td>9.6970</td>
<td>0.3651</td>
<td>-0.3030</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>10</td>
<td>9.9101</td>
<td>0.2308</td>
<td>-0.0899</td>
<td>9.7296</td>
<td>0.2879</td>
<td>-0.2704</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>7.5</td>
<td>7.4680</td>
<td>0.1425</td>
<td>-0.0320</td>
<td>7.3719</td>
<td>0.1587</td>
<td>-0.1281</td>
</tr>
<tr>
<td>$\sigma^2_{11}$</td>
<td>8</td>
<td>7.4288</td>
<td>2.4146</td>
<td>-0.5712</td>
<td>4.6124</td>
<td>1.3881</td>
<td>-3.3876</td>
</tr>
<tr>
<td>$\sigma^2_{12}$</td>
<td>8</td>
<td>7.5921</td>
<td>2.5043</td>
<td>-0.4079</td>
<td>4.8320</td>
<td>1.4949</td>
<td>-3.6180</td>
</tr>
<tr>
<td>$\sigma^2_{13}$</td>
<td>8</td>
<td>7.4709</td>
<td>1.9786</td>
<td>-0.5291</td>
<td>5.4871</td>
<td>3.1748</td>
<td>-2.5129</td>
</tr>
<tr>
<td>$\sigma^2_{21}$</td>
<td>20</td>
<td>20.2472</td>
<td>2.4145</td>
<td>0.2472</td>
<td>21.0195</td>
<td>2.4726</td>
<td>1.0195</td>
</tr>
<tr>
<td>$\sigma^2_{22}$</td>
<td>10</td>
<td>10.2530</td>
<td>1.2182</td>
<td>0.2530</td>
<td>11.3805</td>
<td>1.5084</td>
<td>1.3805</td>
</tr>
<tr>
<td>$\sigma^2_{23}$</td>
<td>3.8</td>
<td>3.9157</td>
<td>0.4575</td>
<td>0.1657</td>
<td>4.5683</td>
<td>0.6626</td>
<td>0.8183</td>
</tr>
</tbody>
</table>

Table A.6. Two-component gamma mixture with $\alpha_1 = (2, 2, 2), \alpha_2 = (5, 10, 15), \beta_1 = (2, 2, 2),$ and $\beta_2 = (2, 1, 0.5)$ for $\lambda = 0.8$. The means and variances are $\alpha \beta$ and $\alpha \beta^2$, respectively. The results are based on 1000 simulations with sample size $n = 300$ each.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM Mean</th>
<th>Std.Error</th>
<th>Bias</th>
<th>Normal Mean</th>
<th>Std.Error</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.8</td>
<td>0.7819</td>
<td>0.0324</td>
<td>-0.0181</td>
<td>0.5818</td>
<td>0.0531</td>
<td>-0.2182</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>4</td>
<td>3.9428</td>
<td>0.1941</td>
<td>-0.0572</td>
<td>3.7175</td>
<td>1.1330</td>
<td>-0.2825</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>4</td>
<td>3.9508</td>
<td>0.2089</td>
<td>-0.0492</td>
<td>3.6957</td>
<td>1.1732</td>
<td>-0.3043</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>4</td>
<td>3.9415</td>
<td>0.2049</td>
<td>-0.0585</td>
<td>3.6579</td>
<td>0.9229</td>
<td>-0.3421</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>10</td>
<td>9.7040</td>
<td>0.6974</td>
<td>-0.2960</td>
<td>7.3773</td>
<td>1.4984</td>
<td>-2.6227</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>10</td>
<td>9.7413</td>
<td>0.5060</td>
<td>-0.2587</td>
<td>7.4495</td>
<td>1.5226</td>
<td>-2.5505</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>7.5</td>
<td>7.4257</td>
<td>0.3346</td>
<td>-0.0743</td>
<td>6.2162</td>
<td>1.1043</td>
<td>-1.2838</td>
</tr>
<tr>
<td>$\sigma^2_{11}$</td>
<td>8</td>
<td>7.5505</td>
<td>1.2515</td>
<td>-0.4495</td>
<td>5.4286</td>
<td>4.6817</td>
<td>-2.5714</td>
</tr>
<tr>
<td>$\sigma^2_{12}$</td>
<td>8</td>
<td>7.5617</td>
<td>1.2490</td>
<td>-0.4383</td>
<td>4.9746</td>
<td>3.6223</td>
<td>-3.0254</td>
</tr>
<tr>
<td>$\sigma^2_{13}$</td>
<td>8</td>
<td>7.6786</td>
<td>1.2004</td>
<td>-0.3214</td>
<td>4.7775</td>
<td>2.1886</td>
<td>-3.2225</td>
</tr>
<tr>
<td>$\sigma^2_{21}$</td>
<td>20</td>
<td>20.4077</td>
<td>4.8353</td>
<td>0.4077</td>
<td>19.9018</td>
<td>6.4602</td>
<td>-0.0982</td>
</tr>
<tr>
<td>$\sigma^2_{22}$</td>
<td>10</td>
<td>11.2074</td>
<td>2.5984</td>
<td>1.2074</td>
<td>14.9890</td>
<td>4.4742</td>
<td>4.9890</td>
</tr>
<tr>
<td>$\sigma^2_{23}$</td>
<td>3.8</td>
<td>4.5688</td>
<td>1.3276</td>
<td>0.8188</td>
<td>9.0715</td>
<td>2.5848</td>
<td>5.3215</td>
</tr>
</tbody>
</table>
A.3 Mixture of Mixed Distributions in 3.2.6

<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>npEM Mean</th>
<th>npEM Std.Error</th>
<th>npEM Bias</th>
<th>Normal Mean</th>
<th>Normal Std.Error</th>
<th>Normal Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.3</td>
<td>0.2953</td>
<td>0.0272</td>
<td>-0.0047</td>
<td>0.3001</td>
<td>0.0287</td>
<td>0.0001</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0.0</td>
<td>-0.0017</td>
<td>0.1222</td>
<td>-0.0017</td>
<td>0.0421</td>
<td>0.1330</td>
<td>0.0421</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>4.0</td>
<td>3.9997</td>
<td>0.2999</td>
<td>-0.0003</td>
<td>3.9969</td>
<td>0.3146</td>
<td>-0.0031</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0.0</td>
<td>-0.1932</td>
<td>0.6442</td>
<td>-0.1932</td>
<td>-0.1046</td>
<td>0.6744</td>
<td>-0.1046</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>2.5</td>
<td>2.4809</td>
<td>0.0793</td>
<td>-0.0191</td>
<td>2.4786</td>
<td>0.0792</td>
<td>-0.0214</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>0.0</td>
<td>0.0319</td>
<td>0.1093</td>
<td>0.0319</td>
<td>0.0075</td>
<td>0.0987</td>
<td>0.0075</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>10.0</td>
<td>9.9824</td>
<td>0.2871</td>
<td>-0.0176</td>
<td>10.0108</td>
<td>0.2924</td>
<td>0.0108</td>
</tr>
<tr>
<td>$\sigma_{11}^2$</td>
<td>1.0</td>
<td>1.0407</td>
<td>0.2181</td>
<td>0.0407</td>
<td>1.0750</td>
<td>0.2025</td>
<td>0.0750</td>
</tr>
<tr>
<td>$\sigma_{12}^2$</td>
<td>8.0</td>
<td>7.8438</td>
<td>1.9217</td>
<td>-0.1562</td>
<td>8.4147</td>
<td>2.1397</td>
<td>0.4147</td>
</tr>
<tr>
<td>$\sigma_{13}^2$</td>
<td>32.0</td>
<td>30.0354</td>
<td>7.4664</td>
<td>-1.9646</td>
<td>30.1853</td>
<td>8.1328</td>
<td>-1.8147</td>
</tr>
<tr>
<td>$\sigma_{21}^2$</td>
<td>1.2</td>
<td>1.2677</td>
<td>0.1603</td>
<td>0.0177</td>
<td>1.3051</td>
<td>0.1733</td>
<td>0.0551</td>
</tr>
<tr>
<td>$\sigma_{22}^2$</td>
<td>2.0</td>
<td>2.1852</td>
<td>0.4772</td>
<td>0.1852</td>
<td>1.8077</td>
<td>0.3249</td>
<td>-0.1923</td>
</tr>
<tr>
<td>$\sigma_{23}^2$</td>
<td>16.0</td>
<td>16.1260</td>
<td>1.9444</td>
<td>0.1260</td>
<td>16.0355</td>
<td>1.7862</td>
<td>0.0355</td>
</tr>
</tbody>
</table>

Table A.7. Two-component mixture with the coordinates within a component having different distributions for $\lambda = 0.3$. The results are based on 1000 simulations with sample size $n = 300$ each.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>True</th>
<th>Mean</th>
<th>Std.Error</th>
<th>Bias</th>
<th>Mean</th>
<th>Std.Error</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>0.8</td>
<td>0.7921</td>
<td>0.0259</td>
<td>-0.0079</td>
<td>0.7932</td>
<td>0.0303</td>
<td>-0.0068</td>
</tr>
<tr>
<td>$\mu_{11}$</td>
<td>0.0</td>
<td>-0.0063</td>
<td>0.0687</td>
<td>-0.0063</td>
<td>0.0043</td>
<td>0.0715</td>
<td>0.0043</td>
</tr>
<tr>
<td>$\mu_{12}$</td>
<td>4.0</td>
<td>3.9809</td>
<td>0.1831</td>
<td>-0.0191</td>
<td>3.9882</td>
<td>0.1942</td>
<td>-0.0118</td>
</tr>
<tr>
<td>$\mu_{13}$</td>
<td>0.0</td>
<td>-0.0737</td>
<td>0.3744</td>
<td>-0.0737</td>
<td>-0.0907</td>
<td>0.3789</td>
<td>-0.0907</td>
</tr>
<tr>
<td>$\mu_{21}$</td>
<td>2.5</td>
<td>2.4267</td>
<td>0.1648</td>
<td>-0.0733</td>
<td>2.4063</td>
<td>0.2013</td>
<td>-0.0937</td>
</tr>
<tr>
<td>$\mu_{22}$</td>
<td>0.0</td>
<td>0.1921</td>
<td>0.2614</td>
<td>0.1921</td>
<td>0.1340</td>
<td>0.2918</td>
<td>0.1340</td>
</tr>
<tr>
<td>$\mu_{23}$</td>
<td>10.0</td>
<td>9.9781</td>
<td>0.5900</td>
<td>-0.0219</td>
<td>10.1036</td>
<td>0.6561</td>
<td>0.1036</td>
</tr>
<tr>
<td>$\sigma^2_{11}$</td>
<td>1.0</td>
<td>0.9937</td>
<td>0.0993</td>
<td>-0.0063</td>
<td>1.0070</td>
<td>0.0998</td>
<td>0.0070</td>
</tr>
<tr>
<td>$\sigma^2_{12}$</td>
<td>8.0</td>
<td>7.8465</td>
<td>1.0970</td>
<td>-0.1535</td>
<td>8.1348</td>
<td>1.1861</td>
<td>0.1348</td>
</tr>
<tr>
<td>$\sigma^2_{13}$</td>
<td>32.0</td>
<td>30.6582</td>
<td>4.7162</td>
<td>-1.3418</td>
<td>29.9540</td>
<td>5.0635</td>
<td>-2.0460</td>
</tr>
<tr>
<td>$\sigma^2_{21}$</td>
<td>1.2</td>
<td>1.3289</td>
<td>0.3427</td>
<td>0.0789</td>
<td>1.4180</td>
<td>0.3992</td>
<td>0.1680</td>
</tr>
<tr>
<td>$\sigma^2_{22}$</td>
<td>2.0</td>
<td>3.4675</td>
<td>2.1494</td>
<td>1.4675</td>
<td>1.9514</td>
<td>1.5686</td>
<td>-0.0486</td>
</tr>
<tr>
<td>$\sigma^2_{23}$</td>
<td>16.0</td>
<td>17.7294</td>
<td>5.5563</td>
<td>1.7294</td>
<td>18.0741</td>
<td>6.4601</td>
<td>2.0741</td>
</tr>
</tbody>
</table>

Table A.8. Two-component mixture with the coordinates within a component having different distributions for $\lambda = 0.8$. The results are based on 1000 simulations with sample size $n = 300$ each.
Appendix B

**mixtools: Short Tutorial for Nonparametric Mixtures**

B.1 Generating and analyzing a multivariate mixture in *mixtools*

Most of the computational techniques used in this thesis are implemented in the R package *mixtools*. In this Appendix we will illustrate how to use some of the most relevant functions in *mixtools* to generate and analyse datasets with the npEM algorithm.

Suppose that we want to generate \( n = 500 \) observations from a two component mixture of trivariate \( (r = 3) \) normals with conditionally independent coordinates. This mixture is written as

\[
g(x) = 0.4\phi(x_1; 0, 1)\phi(x_2; 2, 1)\phi(x_3; 0, 1) + 0.6\phi(x_1; 3, 1)\phi(x_2; 5, 1)\phi(x_3; 3, 1), \quad (B.1)
\]

where \( \phi(\cdot; \mu, \sigma) \) denotes the normal density with mean \( \mu \) and standard deviation \( \sigma \).

The function `rmvnormmix` simulates from a mixture of multivariate uncorrelated normal distributions. The user only need to specify the vector of mixing proportions, and the matrices of component means and standard deviations. In R follow the steps:

```r
R> library(mixtools)
R> n <- 500
R> lambda <- c(0.4, 0.6)
R> mu <- matrix(c(0, 3, 2, 5, 0, 3), 2, 3)
R> sigma <- matrix(1, 2, 3)
R> data <- rmvnormmix(n, lambda, mu, sigma)
```

Notice that `data` is a \( 500 \times 3 \) matrix, where the rows represent the subjects and the columns represent the repeated measures. Now we can apply the `npEM` function to the observation matrix `data` to obtain the results for the npEM algorithm. This function requires the dataset and the number of components (\( m = 0 \)), but there are several
options that can be specified by the user. For example, the blockid option identifies
the coordinates that are assumed to be identically distributed. The default is the most
general model, where a density function is estimated for each component and coordinate.
There is also the samebw option, which if 'TRUE', the same bandwidth for each iteration
and for each component and block is used. If 'FALSE', a separate bandwidth for each
component and block is considered, and the bandwidths are updated at each iteration
of the algorithm. For more details about the npEM function, type “?npEM” in R.

We will start with npEM algorithm for the most general model and allowing
different bandwidths for each component and block.

R> out <- npEM(data, mu0=2, samebw=FALSE)

iteration 1 lambda 0.412 0.588 time 0.057
iteration 2 lambda 0.4082 0.5918 time 0.057
iteration 3 lambda 0.407 0.593 time 0.055
. . . . . . . . .
. . . . . . . . .
. . . . . . . . .
iteration 40 lambda 0.4049 0.5951 time 0.05
iteration 41 lambda 0.4049 0.5951 time 0.051
iteration 42 lambda 0.4049 0.5951 time 0.051
lambda 0.4049 0.5951, total time 2.207 s

The npEM function returns a list of class npEM with the data matrix, the posterior
probabilities, the bandwidths, the blockid structure, the sequence of mixing proportions
over iterations (lambda), the final estimates of the mixing proportions (lambdahat), and
the estimated empirical log-likelihood in each iteration.

R> names(out)

[1] "data"   "posterior" "lambda"   "bandwidth" "blockid"
[6] "lambdahat" "loglik"
In order to see only one element of the list, for example the final estimates of the mixing proportions, just type:

```
R> out$lambda.hat
```

[1] 0.4049206 0.5950794

To summarize the results from `npEM` there are functions such as `summary.npEM` and `print.npEM`. The user can compute the component means and variances (or standard deviations) for each block of coordinates by these commands:

```
R> summary(out)
```

500 observations, 3 coordinates, 2 components, and 3 blocks.

Means (and std. deviations) for each component:

Block #1: Coordinate 1
-0.0145 (1.02) 2.99 (1.03)

Block #2: Coordinate 2
2.1 (0.974) 5.04 (1.02)

Block #3: Coordinate 3
-0.0336 (0.893) 2.99 (0.971)

```
R> summary(out)$means
```

```
component 1 component 2
block 1 -0.01453516 2.987058
block 2 2.10236572 5.037844
block 3 -0.03360655 2.993011
```

```
R> summary(out)$variances
```

```
component 1 component 2
block 1 1.0332843 1.0667838
block 2 0.9492897 1.0327952
block 3 0.7970975 0.9426754
```
To visualize the estimate component density functions we have the `plot.npEM` function. For more details about this function type “?plot.npEM” in R. Here we show the basic commands:

```
R> title <- paste("Does this resemble \( N(\mu[1], \sigma[1]^2) \)
                  and \( N(\mu[2], \sigma[2]^2) \)?", + sep="")
R> par(mfrow=c(1,3))
R> plot(outc, title=title)
```

![Density plots](image)

Figure B.1. Component pdfs for a two component mixture of normals with conditionally independent coordinates.

Notice that we constructed model (B.1) intentionally so that the first and third coordinates are conditionally iid. Thus, we can illustrate how to impose the blocking structure in our model.

```
R> outb <- npEM(data, mu0=2, blockid=c(1, 2, 1), samebw=FALSE)
R> outb$lambdahat
[1] 0.4070444 0.5929556
```

```
R> summary(outb)$means
```

```
component 1 component 2
block 1 -0.02393913  3.000740
block 2  2.11318332  5.040932
```
R> summary(outb)$variances

<table>
<thead>
<tr>
<th></th>
<th>component 1</th>
<th>component 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>block 1</td>
<td>0.9176953</td>
<td>0.9712936</td>
</tr>
<tr>
<td>block 2</td>
<td>0.9667148</td>
<td>1.0337765</td>
</tr>
</tbody>
</table>

Now instead of one plot for each coordinate, we have one plot for each block of conditionally iid coordinates, which in this case there are two.

R> par(mfrow=c(1, 2))
R> plot(outb)

Figure B.2. Component pdfs for a two component mixture of normals with blocks of conditionally iid coordinates.
Bibliography


Vita
TATIANA A. BENAGLIA

EDUCATION

- M.Sc. Statistics, State University of Campinas (UNICAMP), Brazil. April 2004
- B.S. Statistics, State University of Campinas (UNICAMP), Brazil. February 2002

PROFESSIONAL EXPERIENCE

- 2007 - Present: R Developer, co-author of the package mixtools
- 2005 - 2008: Research Assistant, Pennsylvania State University
- 2005: Consultant, Consulting Center, Department of Statistics, Pennsylvania State University
- 2004 - 2008: Teaching Assistant, Pennsylvania State University
- 2004: Statistician, Project - Index Brazil DNA. NEPP, Unicamp, Brazil
- 2000 - 2004: Research Assistant, State University of Campinas, Brazil
- 1999 - 2003: Teaching Assistant, State University of Campinas, Brazil

AWARDS and SCHOLARSHIPS

- 2008: Student Paper Competition in Nonparametric Statistics. ASA
- 2000 - 2002: Undergraduate Research Scholarship. FAPESP
- 2001: 2nd Prize Undergraduate Research Project. Brazilian Statistical Association