



5-2009

# A New Generation of Mixture-Model Cluster Analysis with Information Complexity and the Genetic EM Algorithm

John Andrew Howe

*University of Tennessee - Knoxville*

---

## Recommended Citation

Howe, John Andrew, "A New Generation of Mixture-Model Cluster Analysis with Information Complexity and the Genetic EM Algorithm. " PhD diss., University of Tennessee, 2009.  
[https://trace.tennessee.edu/utk\\_graddiss/863](https://trace.tennessee.edu/utk_graddiss/863)

This Dissertation is brought to you for free and open access by the Graduate School at Trace: Tennessee Research and Creative Exchange. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of Trace: Tennessee Research and Creative Exchange. For more information, please contact [trace@utk.edu](mailto:trace@utk.edu).

To the Graduate Council:

I am submitting herewith a dissertation written by John Andrew Howe entitled "A New Generation of Mixture-Model Cluster Analysis with Information Complexity and the Genetic EM Algorithm." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Administration.

Hamparsum Bozdogan, Major Professor

We have read this dissertation and recommend its acceptance:

Mohammed Mohsin, Adam Petrie, Michael Vose

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

---

To the Graduate Council:

I am submitting herewith a dissertation written by John A. Howe entitled “A New Generation of Mixture-Model Cluster Analysis with Information Complexity and the Genetic EM Algorithm.” I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Business Administration.

Hamparsum Bozdogan, Major Professor

We have read this dissertation  
and recommend its acceptance:

Mohammed Mohsin

Adam Petrie

Michael Vose

Accepted for the Council:

Carolyn R. Hodges  
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

# **A New Generation of Mixture-Model Cluster Analysis with Information Complexity and the Genetic EM Algorithm**

A Dissertation

Presented for the

Doctor of Philosophy Degree

The University of Tennessee, Knoxville

John A. Howe

May 2009

Copyright © 2009 by John A. Howe.  
All rights reserved.

# Dedication

This dissertation is dedicated to my grandfather Denzil Smith, after whom my son John Dennis is named. From him I have learned much about life, and he has given me his curiosity and love of music. As I'm sure he would agree, curiosity and music go with mathematics and logic like spaghetti and meatballs. Thus this dedication seems perfectly reasonable 😊.

# Acknowledgments

I would like to thank all the people who have encouraged and pushed me in my lifelong quest of knowledge and achievement, and specifically those who supported and helped me complete this dissertation.

I want to extend grateful thanks to Paul Castagna, who was instrumental in my meeting Dr. Bozdogan and matriculating with the University of Tennessee. I am deeply indebted to my advisor, Dr. Hamparsum Bozdogan, for expertly guiding my professional development, and imbuing me with his enthusiasm for research.

I would like to thank my doctoral committee members Dr. Mohammed Mohsin, Dr. Adam Petrie, and Dr. Michael Vose for their time and support and useful comments.

Especially, I would like to express my gratitude to my wife and children Karen, John, and Amanda. Without their sacrifice and support, this work would have not been completed.

# Abstract

In this dissertation, we extend several relatively new developments in statistical model selection and data mining in order to improve one of the workhorse statistical tools - mixture modeling (Pearson, 1894). The traditional mixture model assumes data comes from several populations of Gaussian distributions. Thus, what remains is to determine how many distributions, their population parameters, and the mixing proportions. However, real data often do not fit the restrictions of normality very well. It is likely that data from a single population exhibiting either asymmetrical or nonnormal tail behavior could be erroneously modeled as two populations, resulting in suboptimal decisions. To avoid these pitfalls, we develop the mixture model under a broader distributional assumption by fitting a group of multivariate elliptically-contoured distributions (Anderson and Fang, 1990; Fang et al., 1990). Special cases include the multivariate Gaussian and power exponential distributions, as well as the multivariate generalization of the Student's T. This gives us the flexibility to model nonnormal tail and peak behavior, though the symmetry restriction still exists. The literature has many examples of research generalizing the Gaussian mixture model to other distributions (Farrell and Mersereau, 2004; Hasselblad, 1966; John, 1970a), but our effort is more general. Further, we generalize the mixture model to be non-parametric, by developing two types of kernel mixture model. First, we generalize the mixture model to use the truly multivariate kernel density estimators (Wand and Jones, 1995). Additionally, we develop the power exponential product kernel mixture model, which allows the density to adjust to the shape of each dimension independently. Because kernel density estimators enforce no functional form, both of these methods can adapt to nonnormal asymmetric, kurtotic, and tail characteristics.

Over the past two decades or so, evolutionary algorithms have grown in popularity, as they



have provided encouraging results in a variety of optimization problems. Several authors have applied the genetic algorithm - a subset of evolutionary algorithms - to mixture modeling, including Bhuyan et al. (1991), Krishna and Murty (1999), and Wicker (2006). These procedures have the benefit that they bypass computational issues that plague the traditional methods. We extend these initialization and optimization methods by combining them with our updated mixture models. Additionally, we “borrow” results from robust estimation theory (Ledoit and Wolf, 2003; Shurygin, 1983; Thomaz, 2004) in order to data-adaptively regularize population covariance matrices. Numerical instability of the covariance matrix can be a significant problem for mixture modeling, since estimation is typically done on a relatively small subset of the observations. We likewise extend various information criteria (Akaike, 1973; Bozdogan, 1994b; Schwarz, 1978) to the elliptically-contoured and kernel mixture models. Information criteria guide model selection and estimation based on various approximations to the Kullback-Liebler divergence.

Following Bozdogan (1994a), we use these tools to sequentially select the best mixture model, select the best subset of variables, and detect influential observations - *all without making any subjective decisions*. Over the course of this research, we developed a full-featured Matlab toolbox (M<sup>3</sup>) which implements all the new developments in mixture modeling presented in this dissertation. We show results on both simulated and real world datasets.

Keywords: mixture modeling, nonparametric estimation, subset selection, influence detection, evidence-based medical diagnostics, unsupervised classification, robust estimation.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	What is Mixture Modeling? . . . . .	1
1.2	Current Issues in Mixture Modeling . . . . .	2
1.3	Review of Literature . . . . .	2
1.4	Expected Contributions . . . . .	4
1.5	Overview of Thesis . . . . .	4
<b>2</b>	<b>The Gaussian Mixture Model (GMM)</b>	<b>6</b>
2.1	Motivation . . . . .	6
2.2	Initialization - K-Means . . . . .	9
2.3	Optimization - EM . . . . .	11
2.4	Predictive Modeling Using Mixtures . . . . .	13
<b>3</b>	<b>Robust Information Complexity Model Selection Criteria</b>	<b>15</b>
3.1	Definition of Complexity . . . . .	15
3.2	Information Criteria Derived from Kullback-Liebler Divergence . . . . .	16
3.2.1	Model is Correctly Specified . . . . .	18
3.2.2	Bias and Model Misspecification . . . . .	20
3.3	Information Criteria for the GMM . . . . .	23
3.4	Information Criteria for Outlier Detection . . . . .	26
3.5	Robust Covariance Estimation . . . . .	28

<b>4</b>	<b>The Genetic Algorithm (GA)</b>	<b>31</b>
4.1	Structure of the Genetic Algorithm . . . . .	31
4.2	Genetic K-Means . . . . .	38
4.3	Genetic Algorithm for Regularized Mahalanobis Distance . . . . .	40
4.4	Genetic EM Algorithm . . . . .	43
4.5	GA for Subset Selection . . . . .	43
4.6	Why the Genetic Algorithm? . . . . .	45
<b>5</b>	<b>The Symmetric Elliptically-Contoured Mixture Model (ECMM)</b>	<b>50</b>
5.1	Multivariate Symmetric Elliptically-Contoured Distributions . . . . .	50
5.2	Parameter Estimation & Inference . . . . .	51
5.2.1	Parameter Estimation . . . . .	51
5.2.2	Inference . . . . .	54
5.3	Details for EC Subclasses . . . . .	55
5.3.1	Pearson Type II . . . . .	55
5.3.2	Pearson Type VII . . . . .	58
5.3.3	Kotz's Type . . . . .	60
5.4	Hybrid EM Algorithm for the ECMM . . . . .	62
5.5	Information Criteria for the ECMM . . . . .	65
<b>6</b>	<b>The Kernel Density Estimator Mixture Model (KMM)</b>	<b>67</b>
6.1	Kernel Density Estimators . . . . .	67
6.2	Bandwidth Estimation . . . . .	70
6.3	Hybrid EM Algorithm for the KMM . . . . .	75
6.4	Information Criteria for the KMM . . . . .	77
<b>7</b>	<b>The Power Exponential Kernel Mixture Model (PEKMM)</b>	<b>78</b>
7.1	Power Exponential Product Kernel . . . . .	78
7.2	Hybrid EM Algorithm for the PEKMM . . . . .	81
7.3	Information Criteria for the PEKMM . . . . .	82
<b>8</b>	<b>Numerical Results</b>	<b>84</b>
8.1	Traditional GMM . . . . .	85
8.1.1	Simulation S1 - Mixed Overlapping . . . . .	85

8.1.2	Simulation S3 - Spherical Overlapping . . . . .	87
8.1.3	Real data - Iris . . . . .	88
8.1.4	Real data - Aorta . . . . .	88
8.1.5	Real data - Diabetic . . . . .	89
8.1.6	Real data - Cancer . . . . .	90
8.2	Updated GMM . . . . .	92
8.2.1	Simulation S1 - Mixed Overlapping . . . . .	92
8.2.2	Simulation S2 - Ellipsoidal Overlapping . . . . .	93
8.2.3	Simulation S3 - Spherical Overlapping . . . . .	95
8.2.4	Real data - Iris . . . . .	98
8.2.5	Real data - Diabetic . . . . .	101
8.2.6	Real data - Cancer . . . . .	102
8.2.7	Real data - Aorta . . . . .	107
8.3	ECMM . . . . .	111
8.3.1	Simulation S3 - Spherical Overlapping . . . . .	111
8.3.2	Simulation S2 - Ellipsoidal Overlapping . . . . .	113
8.3.3	Simulation S5 - Mixed Overlapping . . . . .	116
8.3.4	Real data - Iris . . . . .	118
8.3.5	Real data - Diabetic . . . . .	119
8.4	KMM . . . . .	122
8.4.1	Simulation S2 - Ellipsoidal Overlapping . . . . .	122
8.4.2	Real data - Diabetic . . . . .	124
8.4.3	Real data - Aorta . . . . .	127
8.5	PEKMM . . . . .	129
8.5.1	Simulation S4 - Nonoverlapping . . . . .	129
8.5.2	Real data - Wine . . . . .	130
8.5.3	Real Data - Colon . . . . .	134
<b>9</b>	<b>Conclusions</b>	<b>138</b>
9.1	Summary of Dissertation . . . . .	138
9.2	Future work . . . . .	139
9.3	Expected Publications . . . . .	140

<b>Bibliography</b>	<b>141</b>
<b>Appendix</b>	<b>151</b>
A.1 MATLAB Toolbox - $M^3$ . . . . .	152
A.2 Datasets . . . . .	161
<b>Vita</b>	<b>176</b>

# List of Tables

2.1	New Datapoints and their Group Classifications. . . . .	14
4.1	Some Univariate Probability Distributions and their Log-likelihoods. . . . .	33
4.2	GA Operational Parameters. . . . .	35
4.3	Complete Enumerative Subset Analysis. . . . .	44
4.4	Various Simulated Annealing Cooling Schedules. . . . .	46
4.5	GA Parameters Varied in First GA Experiment. . . . .	47
4.6	GA Parameters Varied in Second GA Experiment. . . . .	47
6.1	Sample Univariate Kernel Functions. . . . .	69
6.2	Bandwidth Matrix Estimation Methods. . . . .	74
7.1	Maximized Log-likelihoods. . . . .	79
8.1	Simulation S1 - Results from Sole Completed Simulation. . . . .	86
8.2	Simulation S3 - Model Selection Results from Best Simulation With Complete Convergence. . . . .	87
8.3	Iris data - Gaussian Mixture Model Selection Results. . . . .	88
8.4	Aorta data - Results from a Typical Replication. . . . .	89
8.5	Diabetic data - Results from a Typical Replication. . . . .	89
8.6	Cancer data - Confusion Matrix from Best Mixture of Normals Model. . . . .	91
8.7	Simulation S1 - Model Selection Frequencies and Results for GEM(GARM). . . . .	92
8.8	Simulation S1 - Confusion Matrix from Best Simulation. . . . .	92
8.9	Simulation S2 - Results for Mixture of Gaussians Using EM(GKM). . . . .	94

8.10	Simulation S3 - Mixtures of Gaussians with EM(GKM) Results. . . . .	95
8.11	Simulation S3 - Confusion Matrices. . . . .	96
8.12	Simulation S3 - Actual and Estimated Parameters Using Best Model. . . . .	97
8.13	Iris data - Covariance Estimator Selection. . . . .	99
8.14	Iris data - Mixture of Gaussian Models Selected by GEM(GARM). . . . .	99
8.15	Iris data - Results from Best Replication Using <i>ICOMP</i> . . . . .	99
8.16	Iris data - Subset Analysis Using Best $\hat{K} = 3$ Mixture Model. . . . .	100
8.17	Diabetic data - Model Selection Frequencies out of 25 Replications. . . . .	102
8.18	Diabetic data - Summary of Best 5 Replications from the Gaussian Mixture Model. .	102
8.19	Cancer data - Normality Test Results. . . . .	105
8.20	Cancer data - Summary of Best 5 Replications from Mixture of Normals Model. . .	105
8.21	Cancer data - Results from Best Replication. . . . .	105
8.22	Cancer data - Confusion Matrix from Best Mixture of Normals Model. . . . .	105
8.23	Cancer data - Best Five Subset Models Chosen by <i>ICOMP<sub>PEU</sub></i> . . . . .	106
8.24	Cancer data - Confusion Matrix from Best Subset GMM. . . . .	106
8.25	Aorta data - Models Selected by <i>ICOMP<sub>PEU_MISP</sub></i> in All Replications. . . . .	108
8.26	Aorta data - Confusion Matrix from Best Gaussian Mixture Model. . . . .	108
8.27	Aorta data - Partial Post-subset Analysis Results Summary. . . . .	110
8.28	Simulation S3 - ECMM Subclass Selection Results. . . . .	111
8.29	Simulation S3 - Model Selection Frequencies using ECMM with GEM(GARM). . . .	112
8.30	Simulation S3 - Model Scores from Best Pearson type VII Mixture Model Selected by <i>ICOMP</i> . . . . .	112
8.31	Simulation S3 - Confusion Matrices for Best Kotz Type and Pearson Type VII Mix- ture Models. . . . .	112
8.32	Simulation S2 - EC Subclass Selection Results. . . . .	114
8.33	Simulation S2 - Pearson type VII Mixture Model Selection Frequencies. . . . .	114
8.34	Simulation S2 - Pearson type VII Mixture Model Parameter Estimates. . . . .	115
8.35	Simulation S2 - Confusion Matrix from Best ECMM. . . . .	115
8.36	Simulation S5 - Top Five EC Mixture Models from GEM(GARM) by Criteria. . . .	116
8.37	Simulation S5 - Confusion Matrices from Best EC Mixture Models Identified by <i>SBC</i> and <i>ICOMP</i> . . . . .	117
8.38	Iris data - Confusion Matrix from Best Pearson Type VII Mixture Model. . . . .	118

8.39	Diabetic data - Kotz type Mixture Model Selection Frequencies. . . . .	120
8.40	Diabetic data - Best Five Kotz ECMMs Determined by <i>ICOMP<sub>PEU</sub></i> . . . . .	121
8.41	Diabetic data - Confusion Matrix from Best EC Mixture Model. . . . .	121
8.42	Simulation S2 - Model Selection Frequencies for the Kernel Mixture Model. . . . .	122
8.43	Simulation S2 - Results from Best Simulation Using the Kernel Mixture Model. . . . .	123
8.44	Simulation S2 - Confusion Matrix from Best Simulation Using the KMM. . . . .	123
8.45	Simulation S2 - Parameter Estimates from Best Simulation Using the Kernel Mixture Model. . . . .	124
8.46	Diabetic data - Model Selection Frequencies out of Five Replications. . . . .	125
8.47	Diabetic data - Summary of All Replications from the Kernel Mixture Model. . . . .	125
8.48	Diabetic data - Results from Best Mixture of Kernels Replication. . . . .	125
8.49	Diabetic data - Confusion Matrix from Best Replication. . . . .	125
8.50	Diabetic data - Best 10 Subsets from Mixture of Three Kernel Density Estimators Model. . . . .	126
8.51	Aorta data - Results from Fitting Mixtures of Kernels. . . . .	128
8.52	Simulation S4 - Estimated and Actual Shape and Scale Parameters. . . . .	130
8.53	Wine data - Confusion Matrix for Best Model Selected by <i>SBC</i> . . . . .	131
8.54	Wine data - Estimated PE Shape Parameters for Each Dimension in Each Group. . . . .	132
8.55	Wine data - Partial Post-subset Analysis Results Summary. . . . .	132
8.56	Wine data - Classification Rates from Best Subset PE Kernel Mixture models. . . . .	133
8.57	Wine data - Confusion Matrix from Best Subset Mixture Model. . . . .	133
8.58	Colon data - PE Kernel Mixture Model Selection Frequencies from 10 GEM(GARM) replications. . . . .	135
8.59	Colon data - Estimated PE Shape Parameters from Best <i>ICOMP<sub>PEU</sub></i> Model. . . . .	136
8.60	Colon data - Classification Rates from Best Subset PE Kernel Mixture models. . . . .	137
8.61	Colon data - Correlation Matrix. . . . .	137
A.1	Simulation S1 - Data Generation Parameters. . . . .	161
A.2	Simulation S2 - Data Generation Parameters. . . . .	162
A.3	Simulation S3 - Data Generation Parameters. . . . .	163
A.4	Simulation S4 - Data Generation Parameters. . . . .	164
A.5	Simulation S5 - Data Generation Parameters. . . . .	165



A.6 Wine data - Variables. . . . .	174
------------------------------------	-----

# List of Figures

2.1	How Many Groups are There? . . . . .	7
2.2	Demonstration of Mixture of $K = 2$ Gaussians. . . . .	8
2.3	Demonstration of Centroid Selection Algorithm. . . . .	10
2.4	Demonstration of K-Means. . . . .	11
2.5	Mixture of Two Overlapping Gaussians. . . . .	14
3.1	Example of Grouped non-Gaussian Data. . . . .	21
3.2	Demonstrating Influence Detection. . . . .	27
4.1	Final Simulation with Best Fitting Distribution. . . . .	34
4.2	Biased Ranking Bins. . . . .	36
4.3	Demonstrating Stochastic Evolution in Simulated Annealing. . . . .	49
4.4	Distribution of Response Values from Second GA Experiment. . . . .	49
4.5	Factor Profiler from Second GA Experiment. . . . .	49
5.1	Pearson Type II Subclass for $\nu = 0$ . . . . .	56
5.2	Pearson Type II Subclass for $\nu = 1$ . . . . .	57
5.3	Pearson Type VII Subclass for $\nu = 3$ . . . . .	58
5.4	Pearson Type VII Subclass for $\nu = 20$ . . . . .	59
5.5	Kotz Subclass Reduced to PE Reduced to the Laplace. . . . .	60
5.6	Kotz Subclass Reduced to PE Approximating the Uniform. . . . .	61
6.1	Example of Skewed Bivariate Data. . . . .	68
6.2	Demonstrating Kernel Density Estimate Computation. . . . .	69

6.3	Kernel Functions from Table 6.1. . . . .	69
6.4	Histograms of 3 Examples of the Skewed PE Distribution. . . . .	71
6.5	Bivariate KDE Surface and Contours of Skewed Data. . . . .	71
6.6	Demonstrating the Importance of Appropriate Bandwidth Selection. . . . .	72
6.7	Evolution of Likelihood Surface Using KDE Smoothing. . . . .	76
7.1	Fitting PE Univariate Kernel to Each Dimension of Simulation. . . . .	80
8.1	Simulation S1 - Results from Best Model as Selected by Log-likelihood. . . . .	86
8.2	Simulation S3 - Sample Scatter Plot of $X_1$ Against $X_2$ . . . . .	87
8.3	Cancer data - Scatter Plot Matrix of Variables a) Through e). . . . .	90
8.4	Simulation S1 - Scatter Plot of Best Model Structure. . . . .	93
8.5	Simulation S2 - <i>ICOMP</i> and Correct Classification Rate Measurements per Model. . . . .	94
8.6	Simulation S2 - Scatter Plot of Best Model Structure. . . . .	94
8.7	Simulation S3 - Scatter plot of $\hat{K} = 4$ Model from EM(GKM). . . . .	96
8.8	Simulation S3 - Influential Observation Detection Plot of $\hat{K} = 3$ Model from EM(GKM). . . . .	97
8.9	Iris data - Pairwise Scatter Plots. . . . .	99
8.10	Iris data - Influence Detection Plot for $\{2, 4\}$ Subset Model. . . . .	101
8.11	Cancer data - MDS Scatter Plots. . . . .	103
8.12	Cancer data - Detecting Influential Observations. . . . .	106
8.13	Aorta data - Subset Mixture Models with 0% Misclassification. . . . .	109
8.14	Aorta data - Influence Detection for Best Bivariate Subset Model. . . . .	110
8.15	Simulation S3 - Actual Grouping Structure and Structure Estimated by Pearson Type VII Mixture Model. . . . .	113
8.16	Simulation S2 - Progress Plot for GEM Showing Quick Solution Identification. . . . .	115
8.17	Simulation S5 - Actual and Estimated Grouping Structure. . . . .	117
8.18	Demonstrating Slightly Heavier Tails for Bivariate PVII with $\nu = 4$ . . . . .	118
8.19	Diabetic data - MDS Scatter Plots. . . . .	119
8.20	Simulation S2 - Scatter Plot of Best Model Using the Mixture of Kernels. . . . .	123
8.21	Diabetic data - Detecting Influential Observations. . . . .	127
8.22	Simulation S4 - PE kernel Mixture Model Identified by <i>SBC</i> and <i>ICOMP<sub>PEU</sub></i> . . . . .	129
8.23	Wine data - Andrews Curves Plot. . . . .	131
8.24	Wine data - Andrews Curves Plot from Best Subset Mixture Model. . . . .	133

8.25	Colon data - Results from Normality Test. . . . .	134
8.26	Colon data - Two- and Three- Dimensional Scatter Plots of MDS-Reduced Data. . .	135
8.27	Colon data - Identifying Two Possible Influential Observations. . . . .	136
A.1	Simulated Data Format. . . . .	152
A.2	M <sup>3</sup> Toolbox GUI. . . . .	153
A.3	Simulation S1 - Surface and Contour Plots. . . . .	161
A.4	Simulation S2 - Surface and Contour Plots. . . . .	162
A.5	Simulation S3 - Sample Scatter Plot of $X_1$ Against $X_2$ . . . . .	163
A.6	Simulation S4 - Sample Scatter Plot of $X_1$ Against $X_2$ . . . . .	164
A.7	Simulation S5 - Sample Scatter Plot of $X_1$ Against $X_2$ . . . . .	165
A.8	Aorta Data - Demonstrating Nonnormal Characteristics. . . . .	166
A.9	Aorta Data Bivariate Scatter Plots. . . . .	167
A.10	Cancer data - Scatter Plot Matrix of Variables a) Through e). . . . .	169
A.11	Colon data - Grouped Scatterplot Matrix. . . . .	170
A.12	Diabetic Data - Parallel Coordinates Plot. . . . .	171
A.13	Diabetic Data - Scatter Plot Matrix. . . . .	172
A.14	Iris Data - Pairwise Scatter Plots. . . . .	173
A.15	Wine data - Grouped Scatterplot Matrix for $x_1 \dots x_7$ . . . . .	174
A.16	Wine data - Grouped Scatterplot Matrix for $x_8 \dots x_{13}$ . . . . .	175

# Chapter 1

## Introduction

“Oh, [*introduction*]. Jellyman, offspring, offspring jellyman.” - Crush, Finding Nemo

### 1.1 What is Mixture Modeling?

Mixture modeling is a very useful statistical tool - especially when multivariate data is concerned. Consider  $n$  observations on  $p$  measurements from some physical process. A good example is the Fisher iris data - 150 observations of 4 flower characteristics: *petal length*, *petal width*, *sepal length*, and *sepal width*. As any statistician worth his salt could tell you, the iris data contains  $K = 3$  groups; 50 observations each from the varieties *Iris Setosa*, *Iris Versicolor*, and *Iris Virginica*. The problem at hand is to determine that there are three populations from the data itself. The ubiquitous bell curve distribution is usually used in mixture modeling - the researcher fits  $\hat{K}$  distributions to the data. Algorithmically and conceptually, mixture modeling is quite simple:

1. Determine how many populations to fit.
2. Determine initial estimates for group centroids.
3. Utilize the an initialization algorithm to iteratively assign each observation into the closest cluster until convergence.
4. Utilize a second algorithm to further optimize those cluster assignments.

Things, however, are rarely quite that simple.

## 1.2 Current Issues in Mixture Modeling

There are several challenges that confront the researcher wishing to use multivariate mixture modeling. Foremost is the distributional assumption. The distribution of errors discovered by Carl Freidrich Gauss forms the basis for much of statistics. However, this distributional assumption can be too restrictive, and will lead to suboptimal classification of the data in many cases. Perhaps the data exhibit light or heavy tails, is highly peaked, skewed, or some of each.

The requirement of initial centroid estimates by the K-Means algorithm can be someone onerous. For overlapping data, or data of  $p > 2$  dimensions, making *a priori* centroid determinations is no trivial matter. Secondly, it has been demonstrated that the algorithm is not robust to the selection of centroids, leading to different ways to partition the same dataset.

Despite its ubiquity, the typical EM algorithm also has several arguments against its use. It has been characterized as slow to converge (first-order linear convergence), and highly dependant upon initial conditions (which can vary widely from the k-means algorithm). Thirdly, we have observed that the EM algorithm can have difficulty converging to a solution when data exhibit a substantial amount of overlap, or when minimum group sizes are not arbitrarily set. Finally, we claim that maximizing **just** the likelihood, as the EM algorithm does, is suboptimal.

## 1.3 Review of Literature

The Gaussian mixture model (GMM), one of the most mature statistical clustering methods, was first introduced by Pearson (1894). Pearson suggested solving the mixture problem using the method of moments (MoM), requiring nonlinear optimization in high dimensions. Only simple problems were considered until the introduction of computers in the 1960's. Much progress has been made in the last several decades.

The well-known sequential K-Means was first introduced by MacQueen (1967) as a simple and efficient class label initialization scheme. A technical report by Bozdogan (1983) from the University of Illinois at Chicago proposed an intelligent scheme for selecting the center of each hypothesized cluster. In response to accusations regarding the robustness of K-Means solutions, Krishna and Murty (1999) developed a specialized variant of the genetic algorithm (Holland, 1975)

for optimizing the initial group structure - Genetic K-Means (GKM). The Genetic Algorithm with Regularized Mahalanobis distance (GARM), a more general initialization scheme, was developed and extended by Mao and Jain (1996), Song and Shaowei (1997), and Song et al. (1997). For other extensions to the cluster initialization problem, Chen et al. (2004) is an excellent source.

Working with Pearson’s MoM equations, several researchers made simplifying assumptions in order to make the estimation problem more tractable, see Day (1969) and John (1970b). Kabir (1968) introduced and applied a generalized method of moments to the mixture problem; John (1970a) and Rider (1961) relaxed the symmetric distributional assumptions by developing MoM estimators for gamma and exponential univariate mixture models, respectively.

Peters and Walker (1978) introduced a “general iterative procedure” for computing the maximum likelihood estimates for the GMM. Based on the idea of hidden information (the group labels), this came to be called the Expectation-Maximization algorithm. Theoretical considerations of the EM algorithm have been explored by Xu and Jordan (1996) and Ma and Xu (2005), among others. Both Hasselblad (1966) and Redner and Walker (1984) considered the EM algorithm for a mixture of exponential family distributions. In his PhD thesis, Wicker (2006) presented the Genetic EM algorithm (GEM). Finally, Klein and Dubes (1989) and Bandyopadhyay (2005) applied simulated annealing to clustering and fuzzy clustering, respectively.

Bayesian methods for simple mixture problems were considered by Bernardo and Girón (1988). Using the Bayesian framework, model selection considering measurement error was discussed in Pérez and Berger (2002); Diebolt and Robert (1994) used Bayesian sampling methods to estimate mixture parameters. The problem of choosing the number of clusters in a dataset has been discussed in Hartigan (1975) and Marriott (1971). As is well-known, the *likelihood ratio test statistic* (LRT) does not follow a chi-squared distribution in this context. This problem of model selection was further considered when Bozdogan (1981) derived information criteria *AIC* and *SBC* for the Gaussian mixture model. He derived *ICOMP* for the GMM under various covariance structures in Bozdogan (1994b). In the same year, Bozdogan (1994a) utilized information criteria to simultaneously select the number of clusters, subset the variables, and identify influential observations. From the LRT standpoint, several authors including Wolfe (1971); McLachlan (1987); Feng and McCulloch (1994) proposed modifications and / or empirical methods. From the Bayesian framework, re-

cent model selection contributors have been Stephens (2000) and Recharadson and Green (1997). Woo and Sriram (2006) suggested using a density-based minimum Hellinger distance to estimate the number of components. Finally, Wang et al. (2003a,b) considered application of the Gaussian mixture model after translating the data into feature space using reproducing Hilbert-space kernels.

## 1.4 Expected Contributions

In this dissertation, we extend mixture modeling in two primary directions. The first is to relax the distributional assumptions. Whereas previous research in this arena has primarily focused on the Gaussian mixture model, or allowed for other **univariate** distributions, we extend mixture modeling by allowing for more general multivariate distributions. Multivariate symmetric elliptically-contoured (EC) distributions can be used to model varying levels of peakedness or tail behavior. The Gaussian and Laplace distributions are both well-known special cases of the power exponential distribution, which is a special case of Kotz’s type of EC distribution. By varying the probability density generator, the EC class of distributions can be generalized to a host of symmetric probability distributions. Furthermore, we relax all distributional assumptions by introducing the mixture of kernel density estimators (KMM & PEKMM), in which the entire dataset is utilized to compute density estimates. While it is true that kernel density estimation relies on picking an appropriate bandwidth matrix (of which there are many options), we show how to use information criteria to do this without making any subjective decisions. Finally, we augment the mixture model by implementing robust covariance estimators that allow us to partially overcome the “curse of dimensionality”. Using said covariance estimators, we provide an updated regularized Mahalanobis (RM) distance with a more intelligent regularization function. Our RM distance also uses more information about the data dependency structure to scale itself.

## 1.5 Overview of Thesis

The remainder of this dissertation is divided into 8 chapters. In Chapter 2, we present details of the traditional multivariate Gaussian mixture model, including computational algorithms. From here, we move on to justifications of information criteria (IC) and derive various IC for the GMM in Chapter 3. We also show how to using information criteria to identify influential observations in a dataset. Chapter 4 begins with background of the genetic algorithm (GA), then shows how it can be extended to the problem of assigning observations to groups. After presenting details of



GKM, GARM, and GEM, Chapter 4 shows how the GA can be used for the problem of dimension reduction through variable subsetting.

In Chapter 5, we discuss the class of elliptically-contoured distributions and the problem of parameter estimation. We then show the modified EM algorithm and derive various information criteria under these relaxed distributional assumptions. The same topics are presented for the KDE mixture models in Chapters 6 and 7. Numerical results from all these methods are shown in Chapter 8. Finally, Chapter 9 consists of conclusions and suggestions for further future research. All simulated and real datasets are described, with appropriate visualizations, in the appendix.

# Chapter 2

## The Gaussian Mixture Model (GMM)

*“Holds two NCAA division one records; one for most points in a season the other for distance. Former Nick-name: The Mule. The only pro-athlete ever to come out of Collier County and one hell of a **model** American” - Ace Ventura, Ace Ventura, Pet Detective*

### 2.1 Motivation

In the late 1800’s, Karl Pearson (1894) was analyzing a dataset consisting of the ratio of “forehead” breadth to body length for 1000 crabs sampled at Naples by Professor W.F.R. Weldon. He found a mixture of two normal distributions fit the data very well, and concluded that the measurements were made on two separate species of crabs; thus was born mixture modeling. Mixture modeling is a method of partitional cluster analysis (as opposed to hierarchical) in which the researcher pretends to not know the actual group labels for a given dataset. Of course, in real applications, we might not know them, as with Pearson’s crabs. This is in a class of statistical modeling methods called *unsupervised learning*, as opposed to *supervised learning* such as discriminant analysis.

In general, we are given  $X \in \mathbb{R}^{(n \times p)}$  ( $n$  observations and  $p$  measurements), and we want to estimate the number of mixtures (also called clusters/groups/classes/populations) in the data ( $K$ ) and the class identifier for each observation ( $\hat{y}_i \mid X, i = 1 \dots n, \hat{y}_i \in 1 \dots K$ ). A perfect example of this sometimes daunting task is displayed in Figure 2.1. Here we have some bivariate data, and we have created a scatter plot of the first dimension on the x-axis versus the second dimension on the y-axis. Traditionally, mixture modeling, like most of statistics, is based on an assumption of

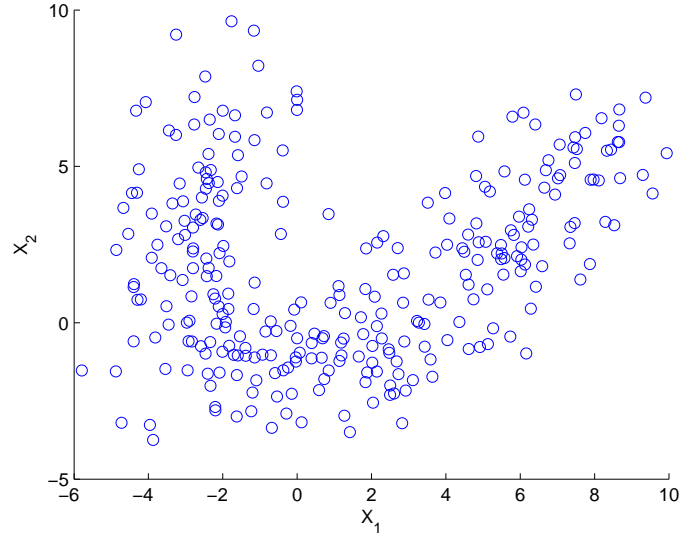


Figure 2.1: How Many Groups are There?

normality.

$$g_k(x_i | \hat{\mu}_k, \hat{\Sigma}_k) = (2\pi)^{-\frac{p}{2}} |\hat{\Sigma}_k|^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (x_i - \hat{\mu}_k)' \hat{\Sigma}_k^{-1} (x_i - \hat{\mu}_k) \right) \quad (2.1)$$

$$\log L(\hat{\theta} | X) = \sum_{i=1}^n \log \left[ \sum_{k=1}^{\hat{K}} \hat{\pi}_k g_k(x_i | \hat{\mu}_k, \hat{\Sigma}_k) \right] \quad (2.2)$$

For a given datapoint and mixture, the multivariate Gaussian probability density  $g_k(x_i | \hat{\mu}_k, \hat{\Sigma}_k)$  is displayed in (2.1), followed by the log-likelihood function for the entire mixture model. For the  $k^{\text{th}}$  population, the mixing proportion, mean vector, and covariance matrix are estimated as shown in (2.4) through (2.6), where

$$I_k(\hat{y}_i) = \begin{cases} 1 & \hat{y}_i = k \\ 0 & \hat{y}_i \neq k \end{cases} \quad (2.3)$$

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^n I_k(\hat{y}_i) \quad (2.4)$$

$$\hat{\mu}_k = \frac{1}{\hat{\pi}_k n} \sum_{i=1}^n x_i I_k(\hat{y}_i) \quad (2.5)$$

$$\hat{\Sigma}_k = \frac{1}{\hat{\pi}_k n} \sum_{i=1}^n [(x_i - \hat{\mu}_k)' (x_i - \hat{\mu}_k)] I_k(\hat{y}_i) \quad (2.6)$$

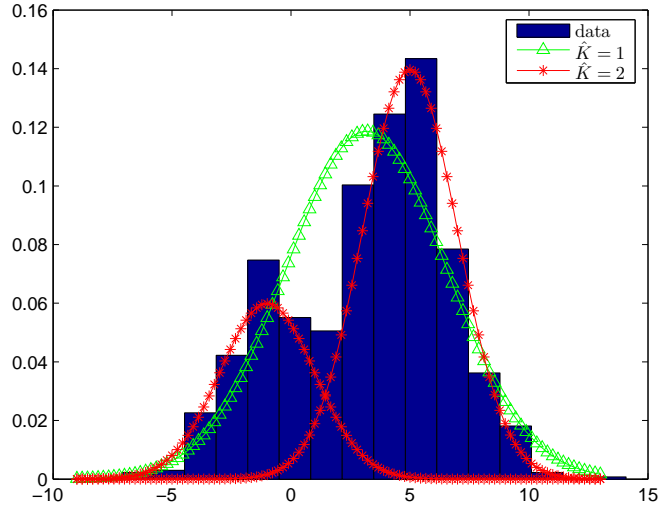


Figure 2.2: Demonstration of Mixture of  $K = 2$  Gaussians.

So as to decrease the computational burden, a “common” covariance matrix is sometimes estimated and applied to all populations, rather than the group-specific (2.6) estimate. We feel this unnecessarily simplifies the problem, and the gain in computation time is probably no longer worth it. A very good visualization of the problem of decomposing a dataset into two distributions can be seen in Figure 2.2. We generated a total of  $n = 1000$  random samples from  $N(\mu = -1, \sigma^2 = 2)$  and  $N(\mu = 5, \sigma^2 = 2)$ , using mixing proportions of  $\pi_1 = 0.3$  and  $\pi_2 = 0.7$ . The green  $\triangle$  curve shows the density estimate computed by fitting a single distribution to the entire dataset. However, the red  $*$  curves, modeling the true mixture, clearly provide a much better fit to the data.

The first step in fitting a GMM is to determine the appropriate number of mixtures  $\hat{K}$  to fit to a dataset, a problem often ignored in much of the literature. Typically, a range is evaluated:  $\hat{K} = 1 \dots K_{\max}$ ; there are several heuristic guidelines for determining  $K_{\max}$  (see Bozdogan, 1994b) including:

- $K_{\max} < \text{ceil}\left(\frac{2n}{(p+1)(p+2)}\right)$
- $K_{\max} \cong \text{ceil}\left(\sqrt{\frac{n}{2}}\right)$
- $K_{\max} = \text{ceil}(\log_2 n)$

Once this first step is complete, we have  $K_{\max}$  different arrangements of the data - we need a method to arbitrate among the results and help us choose the best grouped structure for the data.

A logical choice of criterion is the maximized log-likelihood. Whichever  $\hat{K}$  maximizes the likelihood must fit the data best. In Chapter 3, we'll see a class of model selection criteria that extend and improve upon this.

For each model,  $\hat{K} = 1 \dots K_{\max}$ , the researcher must perform high-dimensional nonlinear optimization of the likelihood function. This is unfortunately due to the complex nature of the mixture problem; there are no closed-form solutions to  $\frac{\partial}{\partial \theta} \log L(\hat{\theta} | X) = 0$ . This is similar to determining MLEs for certain distributions, such as the Weibull, for which the likelihood must be numerically maximized. The Newton-Raphson algorithm could be used for this case, but we need starting values for the shape and scale parameters. As a direct analogue, for numerically optimizing (2.2), we need initial estimates of (2.4) through (2.3); the K-Means algorithm provides these. The numerical optimization of the likelihood is performed by the Expectation Maximization algorithm, first introduced to mixture modeling by Peters and Walker (1978).

## 2.2 Initialization - K-Means

Ironically, the K-Means algorithm, popularized by MacQueen (1967), with the purpose of computing initial parameter values for the EM algorithm requires its own starting values. For the purpose of computing parameter estimates for a mixture of  $\hat{K}$  distributions,  $\hat{K}$  group means are necessary. This *a priori* requirement is one shortcoming of this method. For overlapping data, or data of  $p > 2$  dimensions, appropriate initialization may not be a trivial matter. One approach to centroid selection is the data-adaptive scheme proposed by Bozdogan (1983). This scheme begins by computing the lowest and highest order statistics  $x_{(1)}$  and  $x_{(n)}$ , then follows this procedure:

1. Compute  $\bar{x}_{11} = \frac{x_{(1)} + x_{(n)}}{2}$ ;  $\bar{x}_{11}$  is used as the initial centroid estimate in the case that  $\hat{K} = 1$ .
2. If the researcher is fitting  $\hat{K} = 2$  mixtures, the centroids are  $\hat{\mu}_1 = \bar{x}_{21} = \frac{x_{(1)} + \bar{x}_{11}}{2}$ ,  $\hat{\mu}_2 = \bar{x}_{22} = \frac{\bar{x}_{11} + x_{(n)}}{2}$ .
3. For  $\hat{K} = 3$ , compute  $\bar{x}_{31} = \frac{x_{(1)} + \bar{x}_{21}}{2}$ ,  $\bar{x}_{32} = \frac{\bar{x}_{21} + \bar{x}_{22}}{2}$ ,  $\bar{x}_{33} = \frac{\bar{x}_{22} + x_{(n)}}{2}$ , centroid assignments are  $\hat{\mu}_1 = \bar{x}_{31}$ ,  $\hat{\mu}_2 = \bar{x}_{32}$ ,  $\hat{\mu}_3 = \bar{x}_{33}$ .

This algorithm continues similarly for higher  $\hat{K}$ . As can be seen in the right pane Figure 2.3, the centroid estimates are evenly spaced along a hyperplane through the center of the data. The black \*#-# markers indicate their placements. For example, \*3\_2 shows where the 2<sup>nd</sup> group is estimated

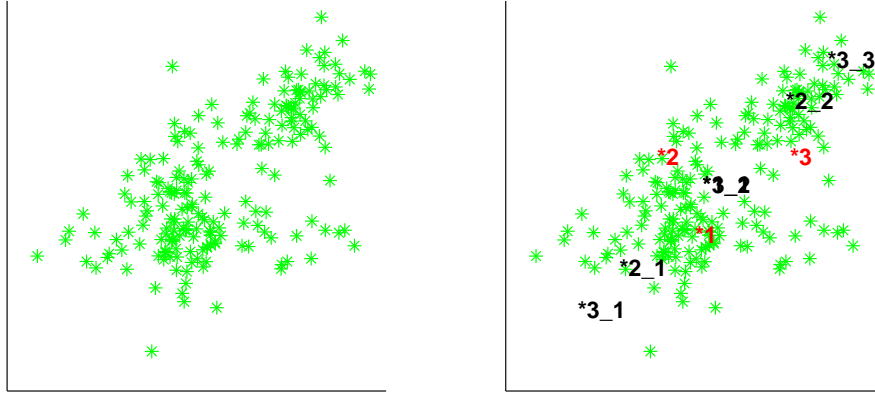


Figure 2.3: Demonstration of Centroid Selection Algorithm.

to be centered, if there are three clusters. The red  $*\#$  indicate the actual centroids used to generate the data. While some of them estimates clearly have some distance over which to migrate, consider the left pane. Where would you center  $\hat{K} = 3$  distributions? Once the initial centroid estimates are computed, the K-Means algorithm alternates assigning datapoints to the nearest cluster, using the Euclidian distance measure (2.7), and recomputing the centroid estimates.

$$e_i(k) = (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)'. \quad (2.7)$$

At the  $t^{\text{th}}$  iteration, the new cluster assignments are first computed, then the centroids are re-estimated:

1.  $\hat{y}_i^{(t+1)} = k$  such that  $e_i(k) = \min_{k=1 \dots \hat{K}} e_i(k)$
2.  $\hat{\mu}_k^{(t+1)} = \sum_{i=1}^n x_i I_k(\hat{y}_i^{(t+1)}) / \sum_{i=1}^n I_k(\hat{y}_i^{(t+1)}), k = 1, \dots, \hat{K}$

At the conclusion of each iteration, the total within-cluster Euclidian distance,

$$E^{(t+1)} = \sum_{i=1}^n \left[ \sum_{k=1}^{\hat{K}} I_k(\hat{y}_i^{(t+1)}) e_i(k) \right], \quad (2.8)$$

is computed; iteration is continued until the absolute difference  $|E^{(t+1)} - E^{(t)}|$  meets some criterion. Hence, the algorithm is a hill-climber. Figure 2.4 graphically represents how this migration improves (2.8). In the left pane, we see 25 circled datapoints in the wrong groups. At some future time step, we suppose these datapoints have migrated into the correct clusters, with a substantial improvement

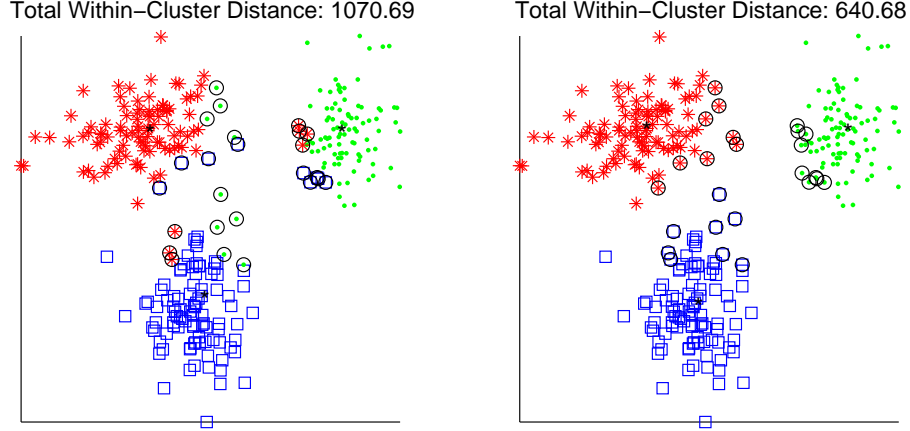


Figure 2.4: Demonstration of K-Means.

in the total distance measure: moving 8.3% of the observations resulted in a 40.2% improvement.

As mentioned in Krishna and Murty (1999), the K-Means algorithm exhibits a strong tendency to converge to suboptimal local minima, and is not robust to the selection of centroids, leading to different ways to partition the same dataset. Other methods for cluster initialization, such as integer programming, dynamic programming, and branch-and-bound methods are known to be computationally intensive, even for a moderate number of observations or mixtures. For further background information regarding clustering algorithms, Jain and Dubes (1989) is a good reference.

## 2.3 Optimization - EM

The expectation maximization algorithm is a fairly general iterative hill-climbing approach for numerical likelihood maximization. The EM algorithm has been derived for a variety of statistical modeling problems as arising from the forms taken by the partial derivatives (Redner and Walker, 1984). However, it seems the most common pedagogical explanation is to approach the problem as did Dempster et al. (1977) - that of considering the data to be incomplete, with the group assignments  $y_i$  missing.

After the estimated mixture assignments are initialized, they are passed on to the EM algorithm, which iterates through alternating **E**xpectation and **M**aximization steps. At the  $t^{\text{th}}$  iteration, the algorithm estimates the posterior probabilities of group membership for datapoint  $i$  and mixture  $k$

using (2.9).

$$\hat{p}_i(k) = \frac{\hat{\pi}_k^{(t-1)} g_k(x_i | \hat{\mu}_k^{(t-1)}, \hat{\Sigma}_k^{(t-1)})}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k^{(t-1)} g_k(x_i | \hat{\mu}_k^{(t-1)}, \hat{\Sigma}_k^{(t-1)})} \quad (2.9)$$

In the subsequent maximization step, estimates for the parameters  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  are recomputed for all populations as shown here:

$$\hat{\pi}_k^{(t)} = \frac{1}{n} \sum_{i=1}^n \hat{p}_i(k), \quad (2.10)$$

$$\hat{\mu}_k^{(t)} = \frac{1}{n \hat{\pi}_k^{(t)}} \sum_{i=1}^n x_i \hat{p}_i(k), \quad (2.11)$$

$$\hat{\Sigma}_k^{(t)} = \frac{1}{n \hat{\pi}_k^{(t)}} \sum_{i=1}^n \hat{p}_i(k) (x_i - \hat{\mu}_k^{(t)})' (x_i - \hat{\mu}_k^{(t)}). \quad (2.12)$$

These steps are iterated until convergence of the log-likelihood,

$$\left| \log L(\hat{\theta} | X)^{(t)} - \log L(\hat{\theta} | X)^{(t-1)} \right| \leq C. \quad (2.13)$$

When the algorithm terminates due to convergence, each observation is assigned to the cluster associated with the highest posterior probability:

$$\hat{y}_i = k \text{ where } \hat{\pi}_k g_k(x_i | \hat{\mu}_k, \hat{\Sigma}_k) = \max_{k=1 \dots \hat{K}} \hat{\pi}_k g_k(x_i | \hat{\mu}_k, \hat{\Sigma}_k). \quad (2.14)$$

This is called the *maximum a posterior* (MAP) rule. The parameter estimates from the final iteration are taken as the maximum likelihood estimators for the Gaussian mixture model.

Despite its popularity and simplicity, the typical EM algorithm has several arguments against its use. The least troublesome is that it has been characterized as having slow convergence rates. Redner and Walker (1984), argue that Newton and quasi-Newton methods should be preferred, considering the slow first-order convergence of the EM. As computation speed increases, and cost decreases, this is less of a problem. More worrisome is the fact that the resultant solution is highly dependent upon the initial estimates. The log-likelihood parameter space is very rugged, especially for large  $p$ , and it is easy for any hill-climber to get stuck in local maxima (Xu and Jordan, 1996) without robust starting values. Of course, we know that the K-Means algorithm is generally incapable of providing robust initialization. Additionally, the traditional EM algorithm can suffer from numerical estimation problems for  $\Sigma_k$ . Without artificially restricting  $n_k$ , a group can become



inestimable when  $\Sigma_k$  becomes ill-conditioned, singular, or non-positive definite. Besides these issues, we have observed that, for mixture modeling, the EM algorithm can get stuck in an oscillating state which will never converge. This can occur when the posterior probability of group membership  $\hat{p}_i(k)$  evolves such that certain datapoints trade back and forth between clusters, or when a mixture has too few datapoints ( $n\hat{\pi}_k^{(t)} \leq p$ ) such that  $\hat{\Sigma}_k^{(t)}$  becomes incomputable. As suggested in Ma and Xu (2005), guaranteed convergence requires that clusters **not be allowed to have an arbitrary size**, and also **restricts the amount of overlap** allowed. We find these restrictions severely limit the practical ability of the EM algorithm. After all, the really interesting and challenging (not to mention real) datasets often exhibit substantial cluster overlap.

## 2.4 Predictive Modeling Using Mixtures

Mixture models provide predictive information through the posterior probability of group membership (2.9). If a practitioner wants to classify a new datapoint into one of  $K$  clusters, it should be assigned to the cluster associated with the highest posterior probability. As an example, consider a simple univariate data set which we generated from a mixture of two Gaussian distributions:

k	Pi	Mu	Stdev	(Actual Parameters)
1	0.38	-3.00	2.00	
2	0.63	3.00	1.00	
k	Pi	Mu	Stdev	(Estimated Parameters)
1	0.37	-3.06	1.87	
2	0.63	2.96	1.04	

We see that the final parameters are quite close to the true values, and the two distributions clearly fit the data well. Using the estimated parameters, we computed the probability of group membership in each distribution for the datapoints

$$X = [-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6].$$

Comparing Table 2.1 to Figure 2.5 suggests the classification rule works well.

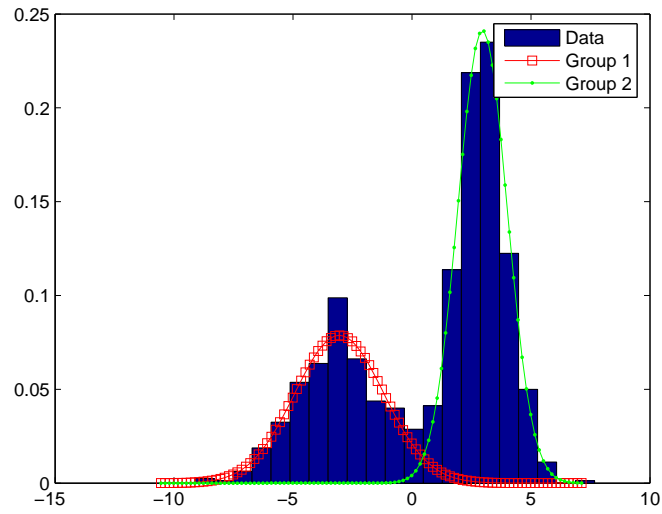


Figure 2.5: Mixture of Two Overlapping Gaussians.

Table 2.1: New Datapoints and their Group Classifications.

$x$	$\pi_1$	$\pi_2$	Group
-4	100.0	0.0	1
-3	100.0	0.0	1
-2	100.0	0.0	1
-1	99.8	0.2	1
0	89.8	10.2	1
1	23.1	76.9	2
2	2.1	97.9	2
3	0.3	99.7	2
4	0.1	99.9	2

# Robust Information Complexity Model

## Selection Criteria

*“You’re a **complex** Freudian hallucination having something to do with my mother, and I don’t know why you have wings, but you have very lovely legs, and you’re a very nice tiny person. . . .”* - Adult Peter Pan, Hook

### 3.1 Definition of Complexity

A reasonable definition of complexity of a p-variate Gaussian distribution, through the covariance matrix  $\Sigma$ , came from Van Emden (1971). Using  $H(x_j)$  to indicate the *marginal entropy* for the  $j^{\text{th}}$  variable and  $H(X)$  to indicate the *joint entropy* for all variables, we have:

$$\begin{aligned}
C_0(\Sigma) &= \sum_{j=1}^p H(x_j) - H(X) \\
&= \frac{1}{2} \sum_{j=1}^p (\log(2\pi) + \log(\sigma_{jj}) + 1) - \frac{1}{2} (p \log(2\pi) + \log|\Sigma| + p) \\
&= \frac{1}{2} \sum_{j=1}^p \log(\sigma_{jj}) - \frac{1}{2} \log|\Sigma|,
\end{aligned} \tag{3.1}$$

where  $\sigma_{jj} = \sigma_j^2$  indicates the variance of the  $j^{\text{th}}$  variable. We use  $tr(\cdot)$  and  $|\cdot|$  to indicate the trace and determinant of a matrix. Some characteristics of  $C_0$  are shown here.

- $C_0(\Sigma) = 0$  iff  $\Sigma$  is a diagonal matrix

- $C_0(\Sigma) = \infty$  iff  $|\Sigma| = 0$
- the first term of (3.1) is not invariant under orthonormal transformations

As a result of this last observation, the *first order maximal information theoretic measure of complexity* of Bozdogan (1988) is generally preferred, shown in (3.2). The maximization is performed over all orthonormal transformations of the coordinate systems  $x_1, x_2, \dots, x_p$ .

$$C_1(\Sigma) = \max_T C_0(\Sigma) = \frac{p}{2} \log \frac{\text{tr}(\Sigma)}{p} - \frac{1}{2} |\Sigma| = \frac{p}{2} \log \frac{\bar{\lambda}_{arith}}{\bar{\lambda}_{geom}} \quad (3.2)$$

Since the covariance matrix measured by  $C_1$  isn't always guaranteed to be of full rank, we would typically replace  $p$  with  $s = \text{rank}(\Sigma)$  in (3.2). Some observations:

- $C_1(\Sigma)$  is the log ratio between the arithmetic and geometric mean of the eigenvalues
- $C_1(\Sigma)$  incorporates the two most basic scalar measures of multivariate scatter - trace and determinant
- $C_1(\Sigma) \rightarrow 0$  as  $\Sigma \rightarrow I_p$
- as interaction between variables increases, so does  $C_1(\Sigma)$

For more details regarding entropic complexity, see Bozdogan (1988).

## 3.2 Information Criteria Derived from Kullback-Liebler Divergence

As previously stated, the first step in mixture modeling is to determine the maximum number of mixtures,  $K_{\max}$  to fit to a dataset. After fitting models  $\hat{K} = 1 \dots K_{\max}$ , we have  $K_{\max}$  different arrangements of the data. Now we need a method to arbitrate among the results and help us choose the best number of groups represented in the data. This is where information criteria come into the picture - the best model for the data is that which minimizes the information criterion (IC) function.

Of all the advances in statistics in the past 50 years, one of the most valuable has been the introduction of elements of information theory. When Akaike (1973) introduced his well-known Akaike's Information Criteria (*AIC*), the subsequent movement introduced a fundamental change in statistical model evaluation problems. Acknowledging the fact that any statistical model is

merely an approximate representation of the data generating process (dgp), information criteria attempt to guide model selection according to Occam's Razor. One restatement of this is

*"Of all possible solutions to a problem, all else equal, the simplest solution is probably the best."* - William of Occam

This mind-set is perhaps the greatest reason for the importance of this advance: for a given dataset, the best model is one which balances a good fit to the data and the desire for parsimony. As model complexity increases, the goodness-of-fit must increase at least as much; otherwise, the additional complexity is not worth the cost. Cost could refer to the actual cost of gathering additional data (variables), but here we mostly refer to the cost of additional estimation uncertainty. Virtually all information criteria penalize a poorly-fitting model with negative twice the maximized log-likelihood, as an asymptotic estimate of the *Kullback-Liebler Information* (KL).

The fundamental basis for all information criteria is the KL divergence (KL distance, KL information,...), first introduced by Kullback and Leibler (1951). The KL distance measures the difference between two probability distributions. Let us denote  $\theta^*$  to be vector of parameters of the true dgp, and  $\theta$  to be any other value of the parameter vector. Let  $f(X | \theta)$  denote the joint density function of  $X$  given  $\theta$ , and let  $f(X | \theta^*)$  indicate the true model. Further, let  $I(\theta^* | \theta)$  denote the KL distance between the true model. Then, since  $x_i, i = 1, 2, \dots, n$  are independent, we have:

$$KL(\theta^*, \theta) = \int_{\mathbb{R}^n} f(X | \theta^*) \log \left[ \frac{f(X | \theta^*)}{f(X | \theta)} \right] dx = \sum_{i=1}^n \int f_i(x_i | \theta^*) \log [f_i(x_i | \theta^*)] dx_i - \sum_{i=1}^n \int f_i(x_i | \theta^*) \log [f_i(x_i | \theta)] dx_i, \quad (3.3)$$

where  $f_i, i = 1, 2, \dots, n$  are the marginal densities of the  $x_i$ . Note that the first term in (3.3) is the usual *negative entropy*  $H(\theta^*, \theta^*) = H(\theta^*)$ , which is constant for a given  $f_i(x_i | \theta^*)$ . The second term is equal to:

$$- \sum_{i=1}^n E_{\theta^*} [\log f_i(x_i | \theta)], \quad (3.4)$$

which can be estimated by

$$- \sum_{i=1}^n \log f_i(x_i | \theta) = -\log L(\theta | X) \quad (3.5)$$

without bias. Note how the true parameter  $\theta^*$  has dropped out of this. Of course,  $\log L(\theta | X)$  is the log likelihood of the observations evaluated at  $\theta$ . In practice, we would estimate the parameter vector, typically using the MLE  $\hat{\theta}$ , and so we use the maximized log likelihood to approximate (3.4).

$$-\sum_{i=1}^n \log f_i(x_i | \hat{\theta}) = -\log L(\hat{\theta} | X) \quad (3.6)$$

Thus, when there are competing models for a dataset, selecting the model with the highest maximized likelihood (or lowest negative maximized likelihood) should provide a model nearest to the true data generating process. All true information criteria use this approximation for the KL distance from the true model to penalize a poorly-fitting model. The difference then, is in the penalty for model complexity.

### 3.2.1 Model is Correctly Specified

The simplest information criterion is *AIC*, which penalizes model complexity with twice the number of estimated parameters,  $m = \text{cardinality}(\theta)$ . Similar to *AIC* is Schwarz's Bayesian Criteria (*SBC*), which penalizes overly-complex models with  $m \log(n)$  (Schwarz, 1978).

$$AIC = -2 \log L(\hat{\theta} | X) + 2m \quad (3.7)$$

$$SBC = -2 \log L(\hat{\theta} | X) + m \log(n) \quad (3.8)$$

*ICOMP*, originally introduced in Bozdogan (1988), is a logical extension of *AIC*. The general form of *ICOMP*

$$ICOMP = -2 \log L(\hat{\theta} | X) + 2C_1(\hat{\mathcal{F}}^{-1}), \quad (3.9)$$

is derived as an approximation to the sum of two KL distances.  $\hat{\mathcal{F}}^{-1}$  indicates the inverse Fisher information matrix (IFIM). We've already seen how the first KL distance is incorporated into *ICOMP* - the maximized likelihood. For the second KL distance, consider that fitting a specific model to a dataset gives rise to an asymptotic covariance matrix

$$Cov(\hat{\theta}) = \Sigma(\hat{\theta}) \quad (3.10)$$

for the MLE  $\hat{\theta}$ . That is,

$$\hat{\theta} \sim N(\theta^*, \Sigma(\hat{\theta}) = \hat{\mathcal{F}}^{-1}). \quad (3.11)$$

The estimated Fisher information matrix is computed as the expectation matrix of the second- and cross- partial derivatives of the maximized log likelihood, as shown in (3.12). Of course, we typically operationalize this in practice by replacing the expected values with the observed values. In fact, there is an extensive body of theory, started by Efron and Hinkley (1978), suggesting that the observed information is a better approximation to the true model covariance matrix.

$$\hat{\mathcal{F}}(\hat{\theta}) = -E_X \left[ \frac{\partial^2}{\partial \theta \partial \theta'} \log(f(X | \hat{\theta})) \right] = -E_X \begin{bmatrix} \frac{\partial^2}{\partial \theta_1^2} & \frac{\partial^2}{\partial \theta_1 \theta_2} & \cdots & \frac{\partial^2}{\partial \theta_1 \theta_m} \\ \frac{\partial^2}{\partial \theta_2 \theta_1} & \frac{\partial^2}{\partial \theta_2^2} & & \\ \vdots & & \ddots & \\ \frac{\partial^2}{\partial \theta_m \theta_1} & \frac{\partial^2}{\partial \theta_m \theta_2} & \cdots & \frac{\partial^2}{\partial \theta_m^2} \end{bmatrix} \quad (3.12)$$

Now invoking the  $C_1(\cdot)$  complexity (3.2) on  $\Sigma(\hat{\theta})$  can be seen as the KL distance between the joint density and the product of marginal densities for a normal random vector with covariance matrix  $\Sigma(\hat{\theta})$ , maximized over all orthonormal transformations of that Gaussian random vector. Hence, using the estimated covariance matrix, we define *ICOMP* as the sum of two Kullback-Liebler distances given by:

$$ICOMP(\hat{\mathcal{F}}^{-1}) = -2 \log L(\hat{\theta} | X) + 2C_1(\hat{\mathcal{F}}^{-1}). \quad (3.13)$$

For more details, see Bozdogan (1990).

A very useful form of *ICOMP* can also be derived as a Bayesian criterion close to maximizing a *posterior expected utility* (PEU), as shown in Bozdogan and Haughton (1998). Here we provide a few highlights of the proof. For a given model  $M$  of dimension  $m$ , we can consider the KL distance between the posterior and prior densities:

$$KL(f_{Post}(\theta | X), f_{Prior}(\theta | M)) = -\frac{m}{2} \log(2\pi) - \frac{m}{2} - \frac{1}{2} \log |\hat{\mathcal{F}}^{-1}| - \log f_{Prior}(\theta | M). \quad (3.14)$$

We can define  $U_1 = KL(f_{Post}(\theta | X), f_{Prior}(\theta | M))$  to be a *utility function* (Lindley, 1956; Poskitt, 1987). Let us define the second utility shown in (3.15).

$$U_2 = \exp \left[ -a \times C_1(\hat{\mathcal{F}}^{-1}) \right] \quad (3.15)$$

For  $a = 1$ , our composite utility is  $U = U_1 \times U_2$ . Applying Poskitt's Corollary 2.2, and employing some regularity conditions, if  $\theta$  lies in our model, the posterior expected utility can be approximated

by

$$\log (PEU) \cong \log f(X \mid \hat{\theta}) + \frac{m}{2} \log (2\pi) + \frac{1}{2} \log |\hat{\mathcal{F}}^{-1}| + \log (U) + \log f_{Prior}(\hat{\theta} \mid M), \quad (3.16)$$

up to order  $O(\frac{1}{n})$  and up to some terms which do not depend on the model  $M$ . Simplifying (3.16) we thus obtain a criterion, to be maximized to choose a model:

$$\log f(X \mid \hat{\theta}) - \frac{m}{2} - C_1(\hat{\mathcal{F}}^{-1}) + \log f(M). \quad (3.17)$$

Of course, with no specific *a priori* information, the prior probability for all models should be equal, so  $f(M)$  can be taken to be a constant term. This gives us  $ICOMP_{PEU}$  which we can minimize to select a best model:

$$ICOMP_{PEU} = -2 \log L(\hat{\theta} \mid X) + m + 2C_1(\hat{\mathcal{F}}^{-1}). \quad (3.18)$$

Note that when we defined the utility

$$U_2 = \exp \left[ -a \times C_1(\hat{\mathcal{F}}^{-1}) \right], \quad (3.19)$$

we considered the constant multiplier  $a$  to be 1 in obtaining the result shown above. Indeed other choices of  $a$  are possible and equally justifiable, giving rise to different penalty functionals. For example, a choice of  $a = \log n$  would yield

$$ICOMP_{PEU} = -2 \log L(\hat{\theta} \mid X) + m + \log(n) C_1(\hat{\mathcal{F}}^{-1}). \quad (3.20)$$

which clearly enforces a stricter penalty. This is the form of  $ICOMP_{PEU}$  we use in this research. Selecting the appropriate penalty for  $ICOMP$  is important to the model selection process. Clearly, the penalty could be arbitrarily set so high so as to force the criteria to always select the simplest model considered.

### 3.2.2 Bias and Model Misspecification

*“Model misspecification is a major, if it is not the dominant, source of error in the quantification of most scientific analysis” - Chatfield (1995).*



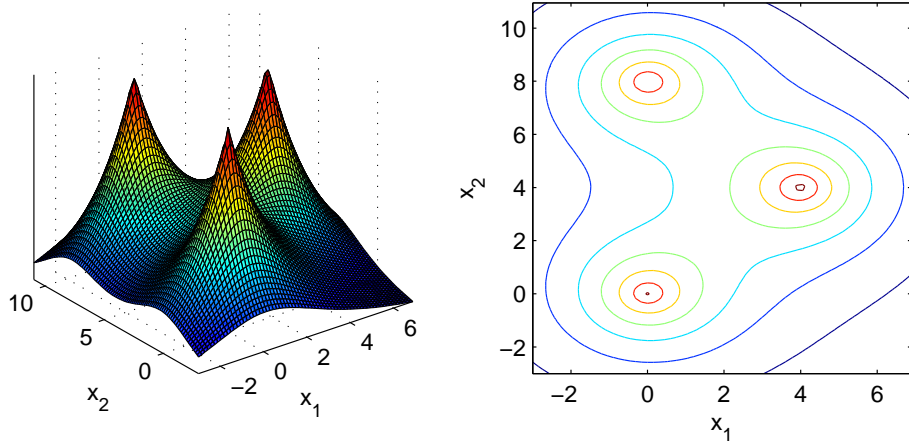


Figure 3.1: Example of Grouped non-Gaussian Data.

*“All models are wrong, but some are useful”* - Box (1979)

In most statistical modeling problems, we can’t assume that the true model is one of those being evaluated; consider assuming a mixture of normals model when the data looks as shown in Figure 3.1, for example. Here we’ve generated surface and contour plots for a mixture of three Laplace distributions, showing heavier non-Gaussian tails. Modeling data like this with a mixture of Gaussian distributions will either lead to lower tail probabilities and/or inflated variance estimates for each group - this can introduce bias into the model. Additionally, it is possible the non-Gaussian tails may be modeled with superfluous groups. When we assume that the true distribution does not belong to the evaluated parametric family of pdfs, that is, if the parameter vector  $\theta^*$  of the distribution is unknown and is estimated by maximizing the likelihood, then it is not any longer true that the average of the maximized log likelihood converges to the expected value of the log likelihood. That is,

$$\frac{1}{n} \log L(\hat{\theta} | X) = \frac{1}{n} \sum_{i=1}^n \log f(x_i | \hat{\theta}) \rightarrow E_X [\log f(X | \hat{\theta})] \quad (3.21)$$

In this case, the bias  $b$  between these two terms is given by

$$b = E_G \left[ \frac{1}{n} \sum_{i=1}^n \log f(x_i | \hat{\theta}) - \int_{\mathbb{R}} \log f(X | \hat{\theta}) dG(X) \right] = \frac{1}{n} \text{tr}(\mathcal{F}^{-1} \mathcal{R}) + O(n^{-2}), \quad (3.22)$$

where the expectation is taken over the true distribution  $G = \prod_{i=1}^n G(x_i)$ . We note that  $\text{tr}(\mathcal{F}^{-1} \mathcal{R})$

is the well known *Lagrange-multiplier test statistic*. Whereas  $\mathcal{F}^{-1}$  is the *inner-product* form of the IFIM,  $\mathcal{R}$  is the *outer-product* form, shown in (3.23).

$$\mathcal{R}(\hat{\theta}) = E \left[ \left( \frac{\partial^2}{\partial \theta \partial \theta'} \log(f(X | \hat{\theta})) \right) \left( \frac{\partial^2}{\partial \theta \partial \theta'} \log(f(X | \hat{\theta})) \right) \right] \quad (3.23)$$

When the model is correctly specified the bias reduces to:

$$b = \frac{1}{n} \text{tr}(\mathcal{F}^{-1} \mathcal{R}) + O(n^{-2}) = \frac{1}{n} \text{tr}(I_m) + O(n^{-2}) = \frac{1}{n} m + O(n^{-2}) \approx \frac{m}{n}. \quad (3.24)$$

Using this approximation, we see how *AIC* is a special case of Takeuchi's information criterion (*TIC*) (see Takeuchi, 1976):

$$TIC = -2 \log L(\hat{\theta} | X) + 2 \text{tr}(\mathcal{F}^{-1} \mathcal{R}) \xrightarrow{\frac{1}{n} \text{tr}(\mathcal{F}^{-1} \mathcal{R}) \approx \frac{m}{n}} AIC = -2 \log L(\hat{\theta} | X) + 2m.$$

Hence, we see explicitly the assumption underlying the penalty employed by *AIC*. See Bozdogan (2000) for more on this.

Using the estimated bias in (3.24), we can further generalize the composite utility shown  $U = U_1 \times U_2$  by

$$U_{MISP} = U_1 \times \exp[2\hat{b}] \times \exp \left[ -\log(n) \times C_1(\hat{\mathcal{F}}^{-1}) \right], \quad (3.25)$$

giving us a form of *ICOMP* that considers the estimated model bias in the penalty:

$$ICOMP_{PEU\_MISP}(\hat{\mathcal{F}}^{-1}) = -2 \log L(\hat{\theta} | X) + m + \frac{2}{n} \text{tr}(\hat{\mathcal{F}}^{-1} \hat{\mathcal{R}}) + \log(n) C_1(\hat{\mathcal{F}}^{-1}). \quad (3.26)$$

**When the true model is not in the model set considered**, which is often the case in practice, simple criteria such as *AIC* will have difficulties identifying the best fitting model, as it does not penalize the presence of *skewness* and *kurtosis*.  $ICOMP_{PEU\_MISP}$ , however, should not suffer these shortcomings.

To summarize: In *ICOMP*, a combination of **lack-of-parsimony** and **profusion-of-complexity** are both penalized by the complexity of the estimated model covariance matrix.  $C_1(\hat{\mathcal{F}}^{-1})$  gives us a scalar measure of the *Cramér-Rao lower bound matrix* (see Cramér, 1946; Rao, 1945, 1947, 1948), taking into account the accuracy of, and relationships between, the parameter estimates. It also

implicitly adjusts for the number of free parameters included in the model, as size of the matrix scales  $m$ . This gives *ICOMP* more power than criteria like *AIC*, which only uses the number of free parameters.

A common question / criticism regarding information criteria relates to the varied penalty functionals. When the criteria identify different models as most appropriate for a dataset, which happens very often, how do we know which criteria to “believe”? However, this criticism should not only be directed towards model selection criteria derived from information-theoretic bases. For example, consider stepwise regression. When considering regressing subsets of independent variables against a dependent variable, criteria such as Mallows Cp (Mallows, 1973),  $R^2$ , and *AIC* are often used. Of course, if all three **always** agreed, we would not need all three. Hence the same question can be directed against non-information-theoretic model selection criteria. At least in the case of information criteria, we could suggest some heuristics:

- only rely upon *AIC* or *SBC* when it is guaranteed that the best approximation to the true model is in the set considered
- use stronger penalties in situations characterized by high-dimensionality and/or highly-parameterized models
- use misspecification-resistant criteria when there is substantial evidence that model assumptions are not met

### 3.3 Information Criteria for the GMM

If we fit the Gaussian mixture model with  $\hat{K}$  groups to a  $p$ -dimensional dataset, the number of parameters is

$$m = \hat{K}p + \hat{K} \frac{p(p+1)}{2} + (\hat{K} - 1); \quad (3.27)$$

$p$  means,  $p(p+1)/2$  unique variances and covariances, and 1 mixing proportion for each cluster. Note that the mixing proportion for the last mixture,  $\hat{\pi}_{\hat{K}}$  is not counted as a parameter to estimate, since the proportions are completely exhaustive. Recall the log-likelihood for the Gaussian mixture model, repeated in (3.28)

$$\log L(\hat{\theta} \mid X) = \sum_{i=1}^n \log \left[ \sum_{k=1}^{\hat{K}} \hat{\pi}_k g_k(x_i \mid \hat{\mu}_k, \hat{\Sigma}_k) \right], \quad (3.28)$$

thus, we have  $AIC$  and  $SBC$  for the GMM in (3.29) and (3.30).

$$AIC = -2 \log L(\hat{\theta} \mid X) + 3m \quad (3.29)$$

$$SBC = -2 \log L(\hat{\theta} \mid X) + \log(n) m \quad (3.30)$$

Note that the penalty for a mixture model is more severe than the usual  $AIC$  (see Bozdogan, 1981). To compute  $ICOMP$ , we need the form of the IFIM for the GMM. Derived in Bozdogan (1994b), we only show the results here. The overall IFIM is

$$\hat{\mathcal{F}}^{-1} = \begin{bmatrix} \hat{\mathcal{F}}_{\pi}^{-1} & & \mathbf{0} \\ & \hat{\mathcal{F}}_1^{-1} & \\ & & \ddots \\ \mathbf{0} & & & \hat{\mathcal{F}}_{\hat{K}}^{-1} \end{bmatrix}. \quad (3.31)$$

The first block of size  $(\hat{K} \times \hat{K})$  (3.32) groups all the mixing proportion estimation variances. The other blocks represent the IFIM for each group.

$$\hat{\mathcal{F}}_{\pi}^{-1} = \begin{bmatrix} \frac{1}{\hat{\pi}_1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \frac{1}{\hat{\pi}_{\hat{K}}} \end{bmatrix} \quad (3.32)$$

Once the parameters for group  $k$  are recovered, they are independent of the subsequent groups, thus we have a block diagonal structure. For the  $k^{\text{th}}$  group, the model covariance matrix is shown in (3.33)

$$\hat{\mathcal{F}}_k^{-1} = \begin{bmatrix} \hat{\Sigma}_k & \mathbf{0} \\ \mathbf{0} & \left(\frac{2}{n_k}\right) D_p^+ (\hat{\Sigma}_k \otimes \hat{\Sigma}_k) D_p^{+'} \end{bmatrix} \quad (3.33)$$

Here,  $\otimes$  is the *Kronecker product*, which multiplies all elements of two matrices. The matrix  $D_p$  is a unique  $\left(p^2 \times \frac{p(p+1)}{2}\right)$  duplication matrix which transforms a square matrix. For example, if  $p = 2$ ,

$$D = \left[ \begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right].$$

$D_p^+$  is its *Moore-Penrose Inverse*:

$$D_p^+ = (D_p' D_p)^{-1} D_p'$$

Obviously, the 2<sup>nd</sup> orthant is of size  $(p \times p)$ , and the 4<sup>th</sup> orthant is of size  $\left(\frac{p(p+1)}{2} \times \frac{p(p+1)}{2}\right)$ . Thus,  $\hat{\mathcal{F}}_k^{-1}$  is a square matrix with dimension of  $\frac{p(p+3)}{2}$ . After some matrix calculus and algebra, *ICOMP* for the mixture of multivariate Gaussians is given in (3.34). Notice that this computation does not require building the entire IFIM, using only traces and determinants of the population covariance matrices.

$$\begin{aligned} ICOMP(\hat{\mathcal{F}}^{-1}) &= -2 \log L(\hat{\theta} \mid X) \\ &+ m \left( \log \left[ \sum_{k=1}^{\hat{K}} \left\{ \frac{tr(\hat{\Sigma}_k)}{\hat{\pi}_k} + \frac{1}{2} \left( tr(\hat{\Sigma}_k^2) + tr(\hat{\Sigma}_k)^2 + 2 \sum_{j=1}^p (\hat{\sigma}_{kjj}^2)^2 \right) \right\} \right] \right. \\ &\left. - \log m \right) - \left\{ (p+2) \sum_{k=1}^{\hat{K}} \log |\hat{\Sigma}_k| - p \sum_{k=1}^{\hat{K}} \log (\hat{\pi}_k n) \right\} - \hat{K} p \log (2n) \end{aligned} \quad (3.34)$$

$(\hat{\sigma}_{kjj}^2)^2$  indicates the square of the  $j^{\text{th}}$  diagonal element of  $\hat{\Sigma}_k$ , and  $m$  is the number of distribution parameters - (3.27) without the  $(\hat{K} - 1)$  term.

For modeling situations characterized by overparameterization,  $ICOMP_{PEU}$  for the GMM is shown in (3.35).

$$ICOMP_{PEU}(\hat{\mathcal{F}}^{-1}) = -2 \log L(\hat{\theta} \mid X) + m + \log(n) C_1(\hat{\mathcal{F}}^{-1}) \quad (3.35)$$

As a practical matter, to compute (3.35), we could simply multiply the *ICOMP* penalty by  $(\log n)/2$ , then add  $m$ .

Finally, in order to correct for bias due to model misspecification,  $ICOMP_{PEU\_MISP}$  uses twice the estimated bias in the penalty term. The limitation here is that analytical computation of  $\hat{\mathcal{R}}$  in the mixture modeling context is currently an intractable problem, so we can't compute  $\hat{b}$  from (3.22). However, it can be shown that the bias can be approximated by  $(nm)/(n - m - 2)$  when the model is in fact misspecified (Bozdogan, 2000). Thus,  $ICOMP_{PEU\_MISP}$  can drive effective model selection with a heavy penalty that directly considers misspecification bias, as shown in

(3.36)

$$ICOMP_{PEU\_MISP}(\hat{\mathcal{F}}^{-1}) = -2 \log L(\hat{\theta} \mid X) + m + \frac{2nm}{n-m-2} + \log(n) C_1(\hat{\mathcal{F}}^{-1}) \quad (3.36)$$

As with  $ICOMP_{PEU}$ , (3.34) is easily adjusted for this different penalty.

### 3.4 Information Criteria for Outlier Detection

As with any statistical modeling procedure the existence of “outliers” can have a substantial impact on results. The typical method for detecting outliers, or influential observations, is to *jackknife* the dataset. If the dataset in question has  $n$  observations, the modeling process is completed  $n$  times, each time leaving out a single observation (hence with  $n-1$  observations in each jackknife replicate). Jackknife is sometimes called the *leave-one-out* method, and is also frequently referred to as *cross-validation*. For each replicate, some metric is computed and compared to the same metric from the entire dataset. By itself, this method is computationally time consuming; when combined with the entire mixture modeling process, it would be take a prohibitive amount of time. Thus, in our implementation, we use the entire dataset (with no cross-validation) to simultaneously identify the number of mixtures and the optimum class assignments. Then, we use the jackknife to fit the best identified mixture model to all  $n$  incomplete datasets. The comparison metric is a ratio based on the information criterion used to identify the optimal mixture model

$$I_i = \frac{SCORE_i}{SCORE}. \quad (3.37)$$

$SCORE_i$  is the value for the dataset missing the  $i^{\text{th}}$  datapoint, and  $SCORE$  is the value for the entire dataset. This ratio should, of course, hover near 1 if there are no influential observations. Finally, to identify observations that seem to have undue influence, we measure the empirical 95% interval of all the ratios, then identify the observations corresponding to ratios outside this range.

If the IC score is positive, a value of  $I_i$  greater than this range leads to a higher IC value for the given mixture model when the  $i^{\text{th}}$  observation is removed. We would interpret this as suggesting the mixture model is positively influenced by this datapoint. Similarly, if  $I_i$  is less than this range, we could possibly improve our mixture model by removing this observation. If the IC score is negative, the opposite conclusions could be made about the  $I_i$  ratios. Figure 3.2 demonstrates this

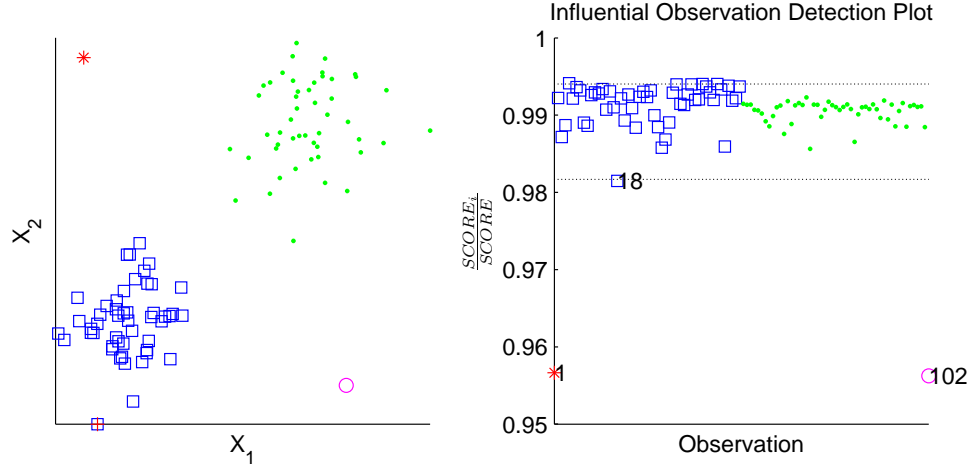


Figure 3.2: Demonstrating Influence Detection.

procedure. Here we've generated a dataset with two groups and two clear outliers - observations  $n_1$  and  $n_{102}$ . We assume a mixture model has been fit to the data that correctly identified the two mixtures, and put each outlier in its own group (this happens). Technically, then, this is the correct structure for this dataset. The output from our procedure shows it identified observations 1, 18, and 102 as being influential and potentially outliers.

```
Full Dataset Score: 766.674
95% Interval of Ratios: [0.982,0.994]
Possible Influential Observation - 1(ICOMP = 733.437, ratio = 0.957)
Possible Influential Observation - 18(ICOMP = 752.475, ratio = 0.981)
Possible Influential Observation - 102(ICOMP = 733.131, ratio = 0.956)
```

Since the *ICOMP* score for each jackknife replicate is lower than that of the entire dataset, we would claim that the mixture model could be improved by removing them (individually, at least). Clearly, removing the two real outliers would dramatically simplify the mixture model - from  $\hat{K} = 4$  to  $\hat{K} = 2$  groups. Note that the 18<sup>th</sup> observation is right on the lower interval limit, suggesting it is not nearly as influential, and probably not an outlier. On the left pane, it is identified by the red +.

This jackknife procedure suffers from two shortcomings. The first is that only the influence of singleton observations is evaluated. Perhaps when taken in triplets, some observations would not be deemed outliers? Of course, this is an issue in any use of the jackknife. The second shortfall, specific to our application, is based on the sequential process. We could probably better evaluate

the influence of the  $i^{\text{th}}$  observation by leaving it out and then fitting the entire modeling process from  $\hat{K} = 1 \dots K_{\text{max}}$ . We have judged that the prohibitive computation time this would require is not justified by the increased accuracy, especially given the subjective nature of outlier detection.

### 3.5 Robust Covariance Estimation

Many fields of research, such as medical imaging and genome science, generate data in which  $p \rightarrow n$ . This can especially be a problem in mixture modeling. While the entire dataset may have many more observations than measurements, it may not be the case that  $n_k \gg p$  for some theorized population. It has long been known that the typical covariance matrix estimate runs into problems for datasets where it is not the case that  $n \gg p$ . First of all, the MLE

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})' (x_i - \hat{\mu}) \quad (3.38)$$

is no longer a good estimate of the true parameter  $\Sigma$ . Secondly, the maximum likelihood estimate becomes ill-conditioned (and not positive definite for  $n \leq p$ ), leading to numerical difficulties in performing the matrix inversion for which  $\hat{\Sigma}$  is usually needed. The traditional response is ridge regularization,

$$\hat{\Sigma}^* = [\hat{\Sigma} + \alpha I_p], \quad (3.39)$$

which works to counteract the instability by adjusting the eigenvalues of  $\hat{\Sigma}$ . The ridge parameter  $\alpha$  is typically chosen to be very small. Of course, this begs the questions

- “How large should  $\alpha$  be?”, and
- “How small can  $\alpha$  be?”.

Now, we can always adjust a matrix to make the inversion numerically stable by adding very large perturbations. However, the information from the data available in the resulting inverted matrix would likely be “washed out” by the perturbations - this is why we must consider the second question here. The answer to ridge regularization questions is to use robust covariance estimators.

Many different robust, or *stoyki*, covariance estimators have been developed as a way to data-adaptively improve ill-conditioned and/or singular covariance matrix estimates. Several of them work by the same mechanism as ridge regularization - perturb the diagonals, and hence, the eigenvalues. Here we’ve only listed the robust covariance estimators we’ve investigated and found to



be useful for mixture modeling. Others performed inconsistently under certain situations involving a relatively high number of groups (Fiebig, 1982; Theil and Fiebig, 1984; Ledoit and Wolf, 2003; Shurygin, 1983).

Maximum Likelihood / Empirical Bayes

$$\hat{\Sigma}_{MLE/EB} = \hat{\Sigma} + \frac{p-1}{(n) \operatorname{tr}(\hat{\Sigma})} I_p \quad (3.40)$$

Stipulated Ridge (Shurygin, 1983)

$$\hat{\Sigma}_{SRE} = \hat{\Sigma} + p(p-1) \left[ (2n) \operatorname{tr}(\hat{\Sigma}) \right]^{-1} I_p \quad (3.41)$$

Convex Sum (Press, 1975; Chen, 1976)

$$\hat{\Sigma}_{CSE} = \frac{n}{n+m} \hat{\Sigma} + \left( 1 - \frac{n}{n+m} \right) \left[ \frac{\operatorname{tr}(\hat{\Sigma})}{p} \right] I_p, \quad m = \frac{\left[ p \left( 1 + \frac{\operatorname{tr}(\hat{\Sigma})^2}{\operatorname{tr}(\hat{\Sigma}^2)} \right) - 2 \right]}{p - \frac{\operatorname{tr}(\hat{\Sigma})^2}{\operatorname{tr}(\hat{\Sigma}^2)}} \quad (3.42)$$

Thomaz Stabilization (Thomaz, 2004)

$$\hat{\Sigma}_{Thomaz} = V \begin{bmatrix} \max(\lambda_1, \bar{\lambda}) & 0 & \dots & 0 \\ 0 & \max(\lambda_2, \bar{\lambda}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \max(\lambda_p, \bar{\lambda}) \end{bmatrix} V, \quad (3.43)$$

$V$  is the matrix formed by the eigenvectors of  $\hat{\Sigma}$

In keeping with the theme of parsimony, we prefer to “monkey around” with the estimated group covariance matrices as little as possible. When a small amount of perturbation is all that is required,  $\hat{\Sigma}_{MLE/EB}$  has a certain appeal. As is clear in (3.40), this is of the same form as the naive ridge regularization, where  $\alpha = (p-1) / \left( (n) \operatorname{tr}(\hat{\Sigma}) \right)$  is determined by the data. In our implementation, the usual covariance estimator is not replaced with one of the robust estimators **every** time it is computed. Within the optimization methods, the covariance matrix  $\hat{\Sigma}_k$  is only smoothed if two measures of matrix condition indicate it to be necessary. The first is the reciprocal of the condition number: if  $\kappa(\hat{\Sigma}_k)^{-1} < 1e^{-10}$ , the matrix is deemed ill-conditioned, which can cause numerical instabilities when it is inverted. The second flag is if  $\hat{\Sigma}_k$  is not positive definite. In

either of these cases, the robust estimator is computed. Sometimes, however, even this approach is unable to compute a usable (invertible) estimator for the true variance-covariance matrix. In these cases, we consider the matrix to be inestimable - this usually causes data points in the  $k^{\text{th}}$  group to migrate out. Note the we are referring explicitly to issues of numerical instability in the inversion process. Mathematically, computation of the scatter matrix can be represented as  $W = M'M$ . If  $W$  has the spectral decomposition noted as above, we have

$$V'WV = V'\lambda V = \lambda\|V\|^2 \geq 0,$$

i.e.,  $W$  should be semi-positive definite. Now, for some small value  $\varepsilon$ , and  $I_p$  indicating the appropriately-sized identity matrix, we would have

$$V'(W + \varepsilon I_p)V = (\lambda + \varepsilon)\|V\|^2 > 0.$$

Hence, ridge regularization in a perfect implementation of linear algebra would result in a semi-definite matrix being transformed to a positive definite invertible matrix. However, the floating-point arithmetic implemented in computers can make this ideal unattainable. Matrix inversion can still suffer numerical instabilities.

Of course, a very good question is “*Which smoothed covariance estimator do I utilize?*”. In order to select the best robust covariance estimator, one which will provide just the necessary regularization (not too much), we use information criteria. We begin by fitting a mixture model of exactly  $K_{\text{max}}$  groups to a given dataset - once for each  $\hat{\Sigma}^*$  in consideration. Whichever covariance estimator produces the minimum score is selected for use with that dataset in the subsequent complete analysis.

# Chapter 4

## The Genetic Algorithm (GA)

*“Don’t you see the danger, uh, John, inherent in what you’re doing here? **Genetic** power’s the most awesome force the planet’s ever seen, but you wield it like a kid that found his dad’s gun.”* - Dr. Malcolm, Jurassic Park

### 4.1 Structure of the Genetic Algorithm

Evolutionary algorithms were studied in the early 1970’s as an alternative to gradient-based optimizers. A typical optimization routine, such as Newton’s method, evolves a single solution to a problem. The evolutionary approach, on the other hand, simulates a large population of potential solutions. These solutions are allowed to interact over time; random mutations and fitness-based selection allow the population to improve, eventually iterating to an optimal solution. Genetic algorithms, a class of evolutionary algorithms, were popularized by Holland (1975) and his students. An article in Scientific American (Holland, 1992) probably contributed to their popularity. Vose (1999) is a good reference for rigorous mathematical bases of the GA.

The genetic algorithm is a stochastic search algorithm that borrows concepts from biological evolution. Biological chromosomes, which determine so much about organisms, are typically represented as binary words - these determine the composition of possible solutions to an optimization problem. For example, consider the problem of selecting the best probability distribution to fit to a given dataset. Say we want to entertain the Students’  $t$  distribution, and maximize the likelihood.

We can use some kind of MoM estimator with the fact that

$$\text{Var}[X] = \frac{\nu}{\nu - 2} \longrightarrow \hat{\nu} = \frac{2S^2}{S^2 - 1}$$

to create a search range, perhaps  $[2, 2\hat{\nu}]$ . The GA would operate on binary strings that, based on the search range, would represent possible values of  $\hat{\nu}$ . To convert real values into binary, we discretize the real range by

$$S = \frac{\max - \min}{2^B - 1}, \quad (4.1)$$

where  $B$  is the number of bits used to encode that value. The number of steps required for the actual value  $R$  are then computed as

$$N = \frac{R - \min}{S}. \quad (4.2)$$

We then encode the value of  $N$  into  $B$  bits. For example, using the  $[2, 2\hat{\nu}]$  range where  $\hat{\nu} = 5$ , 2 would be encoded as a string entirely composed of zeros, and the string of all ones would encode 10. If we used  $B = 32$ , example encodings would be (count right to left)

$$\begin{aligned} 2.05 &= [00000001100110011001100110011001], \\ 5 &= [01011111111111111111111111111111]. \end{aligned}$$

For a simple demonstration of numerical optimization using the GA, we simulated  $M = 10$  sets of random chi-squared data with  $\nu = 12$ . For each set, we used the GA to maximize the likelihood for the eleven distributions shown in Table 4.1. We set the number of generations and populations size both to 50, and the crossover and mutation probabilities 0.75 and 0.10, respectively. Each parameter to estimate was encoded into 32 bits. SBC was used to pick the best fitting model across all simulations. Figure 4.1 shows the histogram of the data from the final simulation with the best fitting distribution overlaid. Clearly, 11.47 is not a bad estimate for 12. In fact, the chi-squared distribution was selected in all simulations. For the problem of assigning datapoints to  $\hat{K}$  clusters, each solution is an  $n$ -length vector of class assignments such that each element can take on any integer in the  $[1, \hat{K}]$  interval. An example with  $x_1, x_4 \in 1$ , and  $x_3, x_5, x_7 \in 3$ , and  $x_2, x_6, x_8 \in 3$  is shown here.

$$\hat{y}_i = \begin{bmatrix} 1 & 3 & 2 & 1 & 2 & 3 & 2 & 3 \end{bmatrix}$$

Table 4.1: Some Univariate Probability Distributions and their Log-likelihoods.

Distribution	log-likelihood / Parameter(s) search range
Cauchy	$-n \log(\pi) - \sum_{i=1}^n \log(1 + (x_i - \theta)^2)$ $\theta \in \left(\bar{X} \pm 2\frac{S}{\sqrt{n}}\right)$
Chi-Squared	$-n \log\left(\Gamma\left(\frac{\nu}{2}\right) 2^{\frac{\nu}{2}}\right) + \left(\frac{\nu}{2} - 1\right) \sum_{i=1}^n \log(x_i) - \frac{1}{2} \sum_{i=1}^n x_i$ $\nu \in \left(0, 2\frac{2\bar{X}}{\sqrt{n}}\right)$
Exponential	$-n \log(b) - \frac{1}{b} \sum_{i=1}^n x_i$ $b \in \left(\bar{X} \pm 2\frac{\bar{X}^2}{\sqrt{n}}\right)$
Gamma	$-n \log(b^a \Gamma(a)) + (a - 1) \sum_{i=1}^n \log(x_i) - \frac{1}{b} \sum_{i=1}^n x_i$ $a \in [0.5, 1.5] \hat{a}, b \in [0.5, 1.5] \hat{b}$
Laplace	Power Exponential with $\beta = \frac{1}{2}$ $\mu \in \left(\bar{X} \pm 2\frac{S}{\sqrt{n}}\right), \sigma \in (0.0001, 1.5S)$
LogNormal	$-\frac{n}{2} \log(2\pi\sigma^2) - \sum_{i=1}^n \log(x_i) - \frac{1}{2\sigma^2} \sum_{i=1}^n (\log(x_i) - \mu)^2$ $\mu \in \left(\bar{X}_{\log} \pm 2\frac{S_{\log}}{\sqrt{n}}\right), \sigma \in (0.0001, 1.5S_{\log})$
Normal	$-\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2$ $\mu \in \left(\bar{X} \pm 2\frac{S}{\sqrt{n}}\right), \sigma \in (0.0001, 1.5S)$
Pareto	$n \log(c) - (c + 1) \sum_{i=1}^n \log(1 + x_i)$ $c \in \left(0, 1.5 \frac{n}{\sum_{i=1}^n \log(1+x_i)}\right)$
Power Exponential	$-n \log\left(\sigma \Gamma\left(1 + \frac{1}{2\beta}\right) 2^{1+\frac{1}{2\beta}}\right) - \frac{1}{2} \sum_{i=1}^n \left \frac{x_i - \mu}{\sigma}\right ^{2\beta}$ $\mu \in \left(\bar{X} \pm 2\frac{S}{\sqrt{n}}\right), \sigma \in (0.0001, 1.5S), \beta \in (0.1, 5)$
Student's t	$n \log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi\nu}\Gamma(\frac{\nu}{2})}\right) - \frac{\nu+1}{2} \sum_{i=1}^n \log\left(1 + \frac{x_i^2}{\nu}\right)$ $\nu \in \left(0, \left\lceil 1.5 \frac{2S^2}{1-S^2} \right\rceil\right)$
Weibull	$n \log(ba^{-b}) + (b - 1) \sum_{i=1}^n \log(x_i) - a^{-b} \sum_{i=1}^n x_i^b$ $a \in \left(0.5\frac{1}{\bar{X}}, 1.5\frac{1}{\bar{X}}\right), b \in \left(0.0001, 1.5\frac{1}{S}\right)$

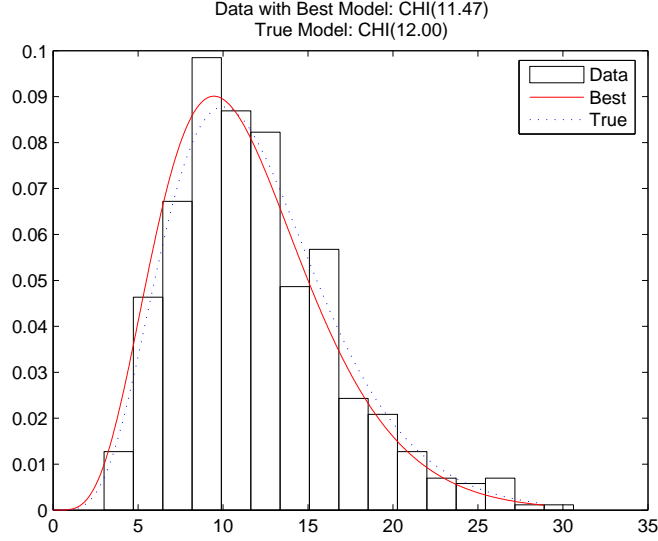


Figure 4.1: Final Simulation with Best Fitting Distribution.

The general procedure in the GA is simple and straightforward:

1. Generate initial population of solutions
2. Score all members of current population
3. Determine how current population interacts to evolve the next generation
4. Mate solutions: perform chromosomal crossover and genetic mutation
5. Pass on offspring to new generation
6. Loop back to step 2 until termination criteria met

There are 8 main operational parameters for the genetic algorithm; sample values used are shown in Table 4.2. What follows is a general description of GA parameters and operators. After providing a good foundation of the fundamental parameters and operators of the genetic algorithm, we will discuss the specific mixture modeling departures. We detail three modified genetic algorithms: two for mixture initialization (GARM and GKM) and one for optimization (GEM).

Table 4.2: GA Operational Parameters.

Parameter	Setting
Number Generations	60
Premature Termination Threshold	40
Population Size	30
Generation Seeding	Ranking
Crossover Probability	0.75
Mutation Probability	0.10
Elitism	On
Objective Function	information criteria

## Number Generations

In the GA, an iteration is typically called a generation, due to the biological conceptualization. Thus, this parameter is fairly self explanatory. There is an important trade-off to note, when selecting the number of generations through which the genetic algorithm will run. More generations mean more computation time; however, not allowing the process to go through enough iterations can mean termination with a suboptimal result.

## Premature Termination Threshold

This parameter is a convergence criteria of the genetic algorithm. If the algorithm has executed a certain number of generations with no improvement in the objective function, we assume that it has converged to an optimal or near-optimal solution. At this point, there's probably not much value in allowing it to continue (though the only cost is computation time, which is cheap). A higher value is better than lower, though the obvious question is "What is high?".

## Population Size

This parameter  $P$  determines how many solutions are evaluated and allowed to interact in each generation. In general, one would expect convergence times to decrease as population size increases, up to a point. After that point, the convergence time increases due to the heavier computation burden per generation. It is difficult to know how large to set this parameter; in fact, there are only heuristic guidelines. For example, in a subsetting problem with  $p$  variables, each generation should evaluate  $P > p$  solutions.

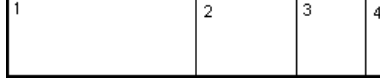


Figure 4.2: Biased Ranking Bins.

## Generation Seeding

From a given population, how do we seed the members of the next generation in preparation for mating? There are three commonly used methods. The simplest would be to randomly pair solutions and mate. In *Tournament Selection*, a set of  $K \ll P$  solutions are uniformly selected and their fitness scores are evaluated. The two best are allowed to mate; this process continues until the next generation is full. Note that a possible benefit of tournament selection is that the objective function is not computed on the entire population. For computationally expensive functions, this can save much time. Conversely, since solutions are randomly selected for evaluation, it would be possible that the best solutions would never be evaluated. The method we use is called *ranking selection*. This would be akin to using a biased roulette wheel, in which the individual bins are of varied size as in Figure 4.2. Bin widths are computed as  $b = (2) / (P(P+1)) \mid b \in [0, 1]$ , then a cumulative sum of these bins is computed. As an example, consider the sorted list of 4 chromosomes - the bin widths are

$$b = \begin{bmatrix} 0.40 & 0.30 & 0.20 & 0.10 \end{bmatrix},$$

so the bin limits are

	1	2	3	4
$B^{low}$	0.00	0.40	0.70	0.90
$B^{upp}$	0.40	0.70	0.90	1.00

Clearly, the larger bins are at the beginning, corresponding to the most fit solutions. At this point,  $P$  random numbers are generated uniformly from  $U \sim [0, 1]$  and placed in the bin such that  $B^{low} \leq U < B^{upp}$ . For each random variate in the  $i^{\text{th}}$  bin, the  $i^{\text{th}}$  solution gets represented in the next generation. In this way, chromosomes with a better objective function value are overrepresented in the mating pool. Finally, the ordering of the solutions is randomly permuted, and solutions are mated sequentially.

## Crossover Probability

There are several ways in which crossover can be implemented. Typical options are



- Single-point (fixed or random)
- Multiple-point (fixed or random)
- Uniform (fixed or random)

We've chosen to use the simplest - randomized single-point crossover. Actually, fixed single-point is arguably simpler, but we have no *a priori* information about where to crossover. For each amorous pair (each of length  $L$ ), a random uniform integer is selected from the range  $[2, L - 1]$ . This range is used, rather than  $[1, L]$ , to protect against probably useless endpoint crossovers - we would prefer to trade the estimated group labels for more than a single observation between two solutions. Their right-most portions are traded starting after this point. For example, if the crossover point is 2, we have

3	2	1	3	3	→	3	2	3	1	2
1	2	3	1	2		1	2	1	3	3

For each mating pair, a random variate from  $U(0, 1)$  is generated; the mating pair undergo the crossover operation if it is less than the crossover probability. Typically, the crossover probability is selected so as to induce frequent crossovers ( $\geq 50\%$ ). If the solutions are not crossed, this procedure just duplicates the original solutions.

## Genetic Mutation

After mating produces offspring, those offspring are then mutated. In a typical GA with binary strings, mutation is simple. One method is to uniformly select solutions from the current population, with probability equal to the mutation rate, to undergo mutation. The probability of mutation must be subjectively chosen; heuristics suggest a small probability ( $\leq 10\%$ ) is generally more appropriate. For each chromosome chosen, elements are then randomly selected (with the same mutation probability) and their bits are switched:  $1 \longleftrightarrow 0$  (a *not* operation). A slightly different approach would be to take all solution vectors in the population as an ensemble, then randomly mutate in a single step. Mutation for mixture modeling is implemented by randomly select elements on the  $\hat{y}_i$  chromosome then reassigning them randomly to different groups. However, we can also utilize problem-specific biased mutation operations that act more intelligently. Mutation is one of the strong points in favor of the GA. Without it, a population of solutions could quickly become homogenous, and get stuck in a local optimum. However, the mutation operator, by inserting different genetic code, can widen the search by allowing a jump to another area of the fitness landscape. To a degree, *simulated annealing* shares this characteristic.

## Objective Function

All optimization or search procedures need some objective function to either maximize or minimize. Newton's method, for example, maximizes a 2<sup>nd</sup> order Taylor series expansion of the likelihood; simulated annealing optimizes the likelihood function itself. The objective function that is best for any GA implementation will depend upon the problem. In our work, we use information criteria to guide the evolution of a solution population.

## Elitism

To protect against the loss of good genetic material (and ensure monotonic improvement in the objective function), the elitism rule is used. After all reproduction operations are complete, this rule copies the most fit chromosome without modification into the next generation. In real life, it can occasionally happen that an especially fit individual will remain a desirable partner for more than one generation; think *Sean Connery*, *Harrison Ford*, or *Charlton Heston*. In the GA, members of a current generation generally die after procreation, leaving the next generation all offspring. However, if the elitism rule is on, the most fit solution from the current generation does not die, but remains to mate with the ensuing generation. Using the elitism rule typically means that population size increases with each generation, which can increase computation time. Other implementations would have the worse population member in the next generation replaced by the current best. We have observed that when roulette selection is used, elitism seems to be of less value.

## 4.2 Genetic K-Means

The genetic algorithm seems to have first been applied to the clustering problem by Bhuyan et al. (1991). Citing concerns that the current efforts were suboptimal, Krishna and Murty (1999) combined the GA and K-Means algorithm. Using finite Markov chain theory, they claimed that GKM (they called it GKA) converges to the global optimum. Like K-Means, GKM seeks to minimize the within-cluster Euclidian distance (2.7) across the entire model. According to the authors, the crossover operator is very expensive, so they did not use one. We find this claim unsupportable, and feel crossover broadens the search, so we use it. Krishna and Murty designed GKA to maximize

a fitness function based on a sigma truncation of the total within-cluster Euclidian distance  $E$ :

$$F(E) = \max(f(E) - (\overline{E} - c\sigma_E), 0), c \in \{1, 2, 3\}. \quad (4.3)$$

$\overline{E}$  and  $\sigma_E$  indicate the sample average and standard deviation of the values of  $E$  in the current population. They seem to provide little justification for this convoluted mechanism, but admit that “*There are many ways of defining such a fitness function.*” Thus, we take the liberty of directly minimizing the total within-cluster Euclidian distance. Not only is this simpler to understand, it works very well. As such, our GKM is slightly different from the original formulation.

For this GA variant, the mutation operator is slightly more complex than the standard mutation operator. The first step is the same - randomly select  $\text{ceil}(\text{mutation probability} \times P)$  offspring solutions to mutate. For each chromosome selected, we uniformly select elements to mutate using the mutation probability. Looping through the selected datapoints, the Euclidian distance from each group is computed and stored. Mutation chances are then computed as

$$E_i(k) = \frac{\max_{k=1 \dots \hat{K}} (e_i(k)) - e_i(k)}{\sum_{k=1}^{\hat{K}} \left[ \max_{k=1 \dots \hat{K}} (e_i(k)) - e_i(k) \right]}, \quad (4.4)$$

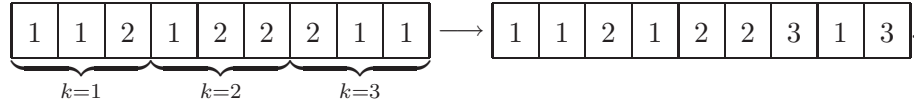
such that the chance for mutating into a given cluster is proportional to the distance from it - the nearer the group, the higher the chances of mutating into it. Though rather obtuse, this formulation is directly from the authors, and we have no explicit reason to use something different. To determine into which cluster, if any, the given datapoint mutates,  $\hat{K}$  uniform random variates are generated, then each is subtracted from  $E_i(k)$ , and the index of the largest positive resulting number is the new value for that datapoint. Along with the mutation and crossover operators, GKM employs a third reproduction operator that can be used to speed convergence. The GKM operation is similar to biased mutation, in that solutions are selected to be modified using the same methodology. By the GKM operator, all elements on selected chromosomes are assigned to the closest group. One characteristic of all three operators (crossover, mutation, GKM) is that they are capable of creating illegal strings where all groups are not represented. For example, we could have

$$\begin{bmatrix} 1 & 3 & 2 & 1 & 2 & 3 & 2 & 3 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 & 2 & 1 & 2 & 2 & 2 & 1 & 1 \end{bmatrix}.$$

This substantially restricts the search of the fitness space. Thus, GKM employs a repair mechanism in which, for each unrepresented cluster,  $p$  datapoints are pseudorandomly assigned into it. Krishna and Murty’s repair operator randomly created a singleton cluster for each missing group. As explicitly defined, this is problematic for two reasons:

- In the case of multiple missing clusters, pure random assignment would allow an observation to be assigned into a missing cluster, then randomly reassigned into a different missing cluster.
- For multivariate data of dimension  $p$ , a singleton group will suffer from  $n < p$ , and the covariance matrix will not be invertible.

Thus, our repair mechanism first divides an illegal string into  $\hat{K}$  disjoint sections. Subsequently, for the  $k^{\text{th}}$  missing group,  $p$  observations in the  $k^{\text{th}}$  section are assigned into it. For example, consider fitting  $\hat{K} = 3$  groups to a bivariate dataset. We could have



### 4.3 Genetic Algorithm for Regularized Mahalanobis Distance

Like GKM, GARM is an intelligent mixture model initialization scheme. One potential shortcoming with both K-Means and GKM is their reliance upon the Euclidian distance. Clustering algorithms based upon the Euclidian distance have an undesirable tendency to split large and elongated clusters; minimizing the Euclidian distance will tend to produce hyperspherical clusters (Mao and Jain, 1996). In general, we can’t assume that data are hyperspherical. Additionally, geometrically speaking, spheres are clearly a subset of ellipsoids - a special case of

$$\frac{(x_1 - c_1)^2}{a_1^2} + \frac{(x_2 - c_2)^2}{a_2^2} + \dots + \frac{(x_p - c_p)^2}{a_p^2} = r^2, \quad (4.5)$$

in which  $a_1 = a_2 = \dots = a_p = 1$ . Therefore, a procedure that is based on an ellipsoidal assumption should be more flexible while still retaining the ability to identify spherical clusters. Mao and Jain (1996) proposed a neural network using the Mahalanobis distance in (4.6) so as to fit hyperellipsoidal clusters.

$$m_i(k) = (x_i - \hat{\mu}_k) \hat{\Sigma}_k^{-1} (x_i - \hat{\mu}_k)' \quad (4.6)$$

Actually, they proposed a *regularized* Mahalanobis (RM) distance so their network could at least partially recover from numerical problems with estimating the within-cluster covariance matrices. Thus, the fitness function for their network was

$$m_i(k) = (x_i - \hat{\mu}_k) \hat{\Sigma}_k^* (x_i - \hat{\mu}_k)', \quad (4.7)$$

where

$$\hat{\Sigma}_k^* = \left[ (1 - \lambda) (\hat{\Sigma}_k + \varepsilon I)^{-1} + \lambda I \right]. \quad (4.8)$$

When  $\lambda = 1$ , (4.7) reduces to the Euclidian distance. For  $0 < \lambda < 1$ , however, (4.7) is a convex combination of the Euclidian and regularized Mahalanobis distances; this allows the clustering procedure to identify differently shaped and oriented clusters. While they argued that  $\lambda$  and  $\varepsilon$  “play a very important role in stabilizing the learning process”, there seems to be no mention of how to select  $\varepsilon$ . They determined that a high value of  $\lambda$  was more important at the beginning of learning than at the end, so it was computed as a decreasing function of the number of iterations in their network

$$\lambda^{(t)} = \max \left\{ \lambda_{\min}, \lambda^{(t-1)} - \Delta\lambda \right\}. \quad (4.9)$$

The starting value of  $\lambda$  is set to 1.0, but  $\lambda_{\min}$  and  $\Delta\lambda$  are user parameters. Thus, effective covariance regularization depends upon three values which must be subjectively set. Though using the regularized Mahalanobis distance is clearly an improvement in generality, it’s not quite the answer. In a critique of the 1996 paper, Song and Shaowei (1997) claimed that the clustering cost function in (4.7) is a constant:  $p(n - 1)$ . They proved that the Mao and Jain (1996) results were not actually caused by their new formulations. Instead, Song et al. (1997) proposed a scaled Mahalanobis Distance given by (4.10).

$$m_i(k) = |\hat{\Sigma}_k|^c (x_i - \hat{\mu}_k) (\hat{\Sigma}_k)^{-1} (x_i - \hat{\mu}_k)' \quad (4.10)$$

Interestingly, their proposal dropped the regularization of the covariance matrix in computing the Mahalanobis distance. The scale parameter  $c$  is constrained to be positive, and they suggest that  $c = 1$  is typically sufficient. Taking cues from Krishna and Murty (1999), they combined their new distance measure with the GA to form what would be called GARM. Whereas GKM minimizes the total within cluster Euclidian distance, GARM minimizes the total within cluster Mahalanobis distance. Like the 1999 paper, they convolute their fitness function with no written justification.

If  $P_p$  is the  $p^{\text{th}}$  member of the current population, its fitness  $f(P_p)$  is computed as

$$f(P_p) = \frac{a(P_p)}{\sum_{p=1}^P a(P_p)}, \quad (4.11)$$

where

$$\begin{aligned} a(P_p) &= \frac{1}{1 + s(P_p)}, \\ s(P_p) &= M(P_p) - \min_{1 \leq p \leq P} M(P_p), \\ M(P_p) &= \sum_{i=1}^n \left[ \sum_{k=1}^{\hat{K}} m_i(k) \right]. \end{aligned}$$

Whereas Song et al. (1997) only used typical GA operators, Wicker (2006) extended GARM to the same level of sophistication as GKM by implementing a biased mutation operator and a special operator called the Mahalanobis operator. The biased mutation operator works just like for GKM, except that mutation chances are computed based on (surprise, surprise), the Mahalanobis distance:

$$M_i(k) = \frac{\max(m_i(k)) - m_i(k)}{\sum_{k=1}^{\hat{K}} [\max(m_i(k)) - m_i(k)]}. \quad (4.12)$$

The Mahalanobis operator similarly uses the Mahalanobis distances to move each datapoint into the closest population, for randomly selected solutions. Additionally, he pointed out that an appealing value for the scale parameter is  $c = \frac{1}{2}$ , since  $|\hat{\Sigma}_k|^{\frac{1}{2}}$  is the square root of the generalized variance for mixture  $k$ .

We propose two further modifications to the regularized Mahalanobis distance.

$$m_i(k) = C_1 (\hat{\Sigma}_k^*) (x_i - \hat{\mu}_k) (\hat{\Sigma}_k^*)^{-1} (x_i - \hat{\mu}_k)' \quad (4.13)$$

The first is to regularize the estimated covariance matrix using one of the robust covariance estimators shown in (3.40) through (3.43). Besides the fact that there are datasets for which ridge regularization is insufficient, it prevents us from having to subjectively choose values for  $\lambda$  and  $\varepsilon$ . Secondly, we scale the RM distance by the first order complexity measure of the covariance matrix,

repeated in (4.14).

$$C_1(\hat{\Sigma}_k^*) = \frac{s}{2} \log \left( \frac{\text{tr}(\hat{\Sigma}_k^*)}{s} \right) - \frac{1}{2} \log |\hat{\Sigma}_k^*|, s = \text{rank}(\hat{\Sigma}_k^*) \quad (4.14)$$

This is advantageous because we no longer have to choose a value of  $c$ . Even more valuable is the fact that the  $C_1$  measure considers both the trace **and** the determinant.

## 4.4 Genetic EM Algorithm

The final specialized GA to be discussed is the genetic expectation maximization algorithm GEM, introduced by Wicker (2006). As already discussed, the EM algorithm operates directly on the log-likelihood with the goal of finding a maximum. Due to the ruggedness of the parameter landscape, it has a tendency to get stuck at local maxima that may be suboptimal, when it converges at all. The GEM algorithm uses a modified GA to search the parameter space more intelligently while operating on chromosomes representing the estimated class labels  $\hat{y}_i$ . While Wicker introduced GEM to search the likelihood space, we extend it further by allowing GEM to optimize the information criteria functions already discussed. Analogous to GARM and GKM, GEM uses a biased mutation operator. The implementation is exactly the same as the others with the exception that probability of group membership  $\hat{p}_i(k)$  (see (2.9), for example) for each cluster is utilized. Mutation chances are then computed as

$$\hat{P}_i(k) = \frac{\max_{k=1 \dots \hat{K}} (\hat{p}_i(k)) - \hat{p}_i(k)}{\sum_{k=1}^{\hat{K}} \left[ \max_{k=1 \dots \hat{K}} (\hat{p}_i(k)) - \hat{p}_i(k) \right]}, \quad (4.15)$$

such that the chance for mutating into the  $i^{\text{th}}$  cluster is directly proportional to the probability. Likewise, GEM has its own operator called the posterior operator in which all datapoints in selected chromosomes are mutated into the mixture to which they're most likely to belong.

## 4.5 GA for Subset Selection

A common theme in statistical modeling is dimension reduction. For example, in analyzing a designed experiment, we like to identify a subset of the tested factors that has an effect on the response. When variables in a dataset share some information (are colinear), the data are often mapped into a smaller set of orthogonal variables called principle components (PCs). Each se-

Table 4.3: Complete Enumerative Subset Analysis.

$p$	$2^p - 1$
5	31
10	1,023
20	1,048,575
50	1,125,899,906,842,623

quential PC is designed to account for less of the variability in the original data. However, this process can make prediction and/or interpretation of the results difficult, especially for classification procedures. In addition, PC analysis is based on the total variability in a dataset, but cluster analysis is concerned with total *within group* variability. Thus, translating the original data into orthogonal PCs erases the group structure. Consider a dataset with  $p = 15$  variables; there is probably some cost associated with making each measurement. Perhaps we would like to identify a model that fits the data well and/or makes good predictions while only using  $p = 5$  variables, for example. Similarly, in mixture model cluster analysis, we would like to determine which variables in the data give the best separation among the clusters; Bozdogan (1994a) introduced this concept using complete enumeration. Since there are  $2^p - 1$  nontrivial subsets of the variables in a dataset, this scales very poorly, as demonstrated in Table 4.3. After an optimal mixture model is found for a dataset, we use the GA to identify a subset of the original variables that improves upon the information criterion score of the saturated (all variables) mixture model. For this application, the GA is a  $p$ -length binary vector where each entry identifies whether a specific variable is included (1) or excluded (0). For example, consider a dataset with  $p = 7$  dimensions. The chromosome shown below would exclude the second, third, and sixth variables.

$$[1001101] \longrightarrow X^* = \{x_1, x_4, x_5, x_7\}.$$

For each subset  $X^*$  evaluated, the identified mixture model is fit to  $X^*$  and the appropriate information criterion is scored. Subsets corresponding to lower scores potentially discriminate between the groups in the data better than the full dataset.



## 4.6 Why the Genetic Algorithm?

Most of the likelihood optimization methods developed for mixture modeling are gradient followers. The two primary reasons for not using these methods are as already mentioned:

- extreme sensitivity to initial values
- high tendency to converge to local, not global, optima

Additionally, many algorithms exhibit slow convergence rates unless started very near the global optima. Specifically for the traditional EM algorithm, guaranteed convergence requires unrealistic restrictions regarding cluster overlap and size. One feature of gradient following algorithms is that they are not allowed to make a “bad” move. However, depending upon the function to optimize and the initial values, a certain number of apparently bad moves may be required to get the algorithm heading in the correct direction. Conjugate gradient methods could be used, but they typically require the first and/or second derivatives of the likelihood to be available *analytically*. This can be quite costly, even when the derivatives are actually available. These derivatives are not, in fact, analytically calculable for the mixture model - recall in Chapter 3 that we used the *observed* Fisher information matrix because of this. Additionally, conjugate gradient methods are also constrained to only move in the optimal direction. Simulated annealing (SA), a stochastic gradient follower that allows bad moves, was first applied to the clustering problem by Klein and Dubes (1989).

Biology gave us the GA, and metallurgy gave us SA. The way simulated annealing optimizes a function is similar to the way the cooling of a molten metal is controlled so as to increase the size and homogeneity of its crystals while decreasing the energy in the lattice. The latent heat liberates the atoms from their local minima and allows them to wander randomly through states of higher energy; the controlled cooling gives the material more chances of finding configurations with lower internal energy. The function to minimize represents the internal energy. The general algorithm is:

1. Set initial state and temperature, compute initial energy.
2. Update temperature, find neighboring state to consider, and compute its energy.
3. If energy is lower, move current state. If energy is higher, move current state with some probability, dependent upon the cost and current temperature.
4. Update time step, then loop back to step 2 until termination criteria met.

Table 4.4: Various Simulated Annealing Cooling Schedules.

Type	Cooling Schedule
Adaptive	$T_t = \exp\left(-ct^{\frac{1}{p}}\right)$
Cauchy	$T_t = \frac{T_0}{t}$
Boltzman	$T_t = \frac{T_0}{\log(t)}$

Just like Linux or Baskin Robbins<sup>TM</sup>, there are various “flavors” of simulated annealing, shown in Table 4.4, as determined by the cooling schedule. For our problem, SA would optimize the mixture likelihood function ((2.2) for example). Selection of  $T_0$  is crucial to success of the procedure, as it is used in computing the probability with which the current state is moved to one with higher energy. The way SA is designed, as the temperature decreases, the probability of making a bad move goes to 0. The idea is that when the temperature is high, we want to search enough of the state space so that as cooling occurs, we end up near a global optimum. Figure 4.3 demonstrates how states evolve for a maximization problem. If the neighbor function selects the point  $\theta_{gt+1}$  (gt indicating good), SA is guaranteed to move there. However, if the neighbor is  $\theta_{bt+1}$  (bt indicating bad), SA will move there with a certain probability that is a function of how long the algorithm has been running. Simulated annealing is highly dependent upon three subjective decisions for which there seem to be few guidelines in the literature.

- the value of  $T_0$
- the cooling schedule (adaptive, Cauchy, Boltzman, etc. . .)
- the neighbor function

It is undeniably true that there are also several parameters to be subjectively set when using the GA, with heuristics available only for some of them. However, in two designed experiments, we evaluated the sensitivity of the GA’s performance to parameter values. Both experiments suggest the algorithm is fairly robust.

For the first experiment, we varied the parameters shown in Table 4.5. The context for this experiment was multivariate subsetting using cable television market segmentation data of Anderson and Steen (1994), with  $p = 7$  and  $n = 101$ . There are only  $2^7 - 1 = 127$  nontrivial subsets, so we could perform complete enumerative analysis on this dataset to determine the optimal subset of variables for comparison. For this experiment, the response was the frequency, across all generations, with which the procedure selected the known optimal subset. Clearly, a higher

Table 4.5: GA Parameters Varied in First GA Experiment.

Parameter	Low	High
Number Generations	50	100
Population Size	10	30
Crossover Probability	0.60	0.90
Mutation Probability	0.025	0.10
Elitism	Off	On

Table 4.6: GA Parameters Varied in Second GA Experiment.

Parameter	Low	High
Number Generations	50	100
Premature Termination Threshold	15	45
Population Size	25	100
Generation Seeding	Random	Ranking
Crossover Probability	0.45	0.75
Mutation Probability	0.05	0.25
Mahalanobis Operation	Off	On

frequency indicates superior performance of the GA. We used a resolution IV design model as an exploratory step. Analysis of the results suggested possible confounding between certain higher-order effects and population size, so we folded the design. Numerical analysis of the final results showed population size was the only significant effect, though it only explained half of the variation in the response:  $R_{adj}^2 = 0.5057$ .

In a second designed experiment - a full factorial design model - the performance of GARM, as measured by a function of  $\hat{y}_i = y_i$ , was evaluated as parameters were varied, shown in Table 4.6. Of course, having more estimated class labels matching the actual class labels would indicate superior performance. For this experiment, our data was generated using a simulation protocol similar to S1, with  $n_k = 85$  observations in each group. Figure 4.4 demonstrates a clear bimodal shape in the response values - thus our parameter ranges were wide enough to induce some difference. In this experiment, the most significant factor was whether or not the Mahalanobis operator was used, accounting for almost 70% of the variability in the response. Six other factors were shown to have a statistically significant but unsubstantial effect on the performance of the GA, as shown in Figure 4.5. In fact, the other six factors only explained 15% of the variability in the response, collectively.

Of course, as the Mahalanobis operation is an integral part of GARM, there'd be no reason not to use it. Thus, in reality we see that this experiment found no substantial sensitivity to parameter values chosen. In summary, these two experiments suggest that the genetic algorithm, as used here, is relatively robust to the actual parameter values chosen. Of course, this assumes intelligent values are chosen.

Genetic algorithms have been used for statistical modeling problems such as Bayesian sampling (Liang and Wong, 2001), robust regression (Burns, 1992), and experimental design (Hamada et al., 2001). Meyer (2003) developed a GA for maximization with linear equality constraints and bounded solutions, then applied it to robust regression and density estimation. Successful application of the GA has been reported in fields such as finance (Neely et al., 1997), econometrics (Routledge, 1999), gaming (West and Linster, 2003), and image processing (Bhandarkar et al., 1994). However, evolutionary algorithms are in a class called *deterministic non-repeating black box search algorithms* (DNBBSA); by the “No Free Lunch” theorem (Wolpert and Macready, 1997) it is true that all DNBBSAs perform equally well **on average**. However, the superior performance of the GA for difficult mixture modeling cluster analysis and variable subsetting problems is well established (and extended here). Additionally, according to personal correspondence with Dr. Micheal Vose,

*“Nothing beats enumeration on average, \*unless\* domain specific knowledge is used.*

*That is roughly the message of the “No Free Lunch” theorem.”*

We suggest that GKM, GARM, and GEM all combine the GA with domain knowledge, which is why they work so well.

Finally, there are incontrovertibly other stochastic or automata-based algorithms that could be considered. Examples include Artificial Neural Networks (ANN), the Artificial Bee Colony (ABC) optimization algorithm, or the Touring Ant Colony Optimization (TACO) algorithm.

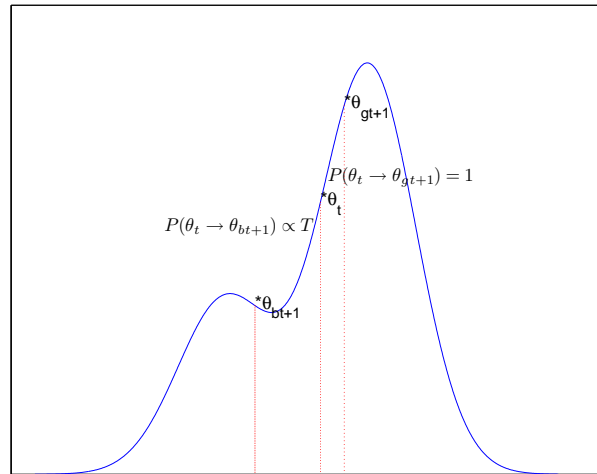


Figure 4.3: Demonstrating Stochastic Evolution in Simulated Annealing.

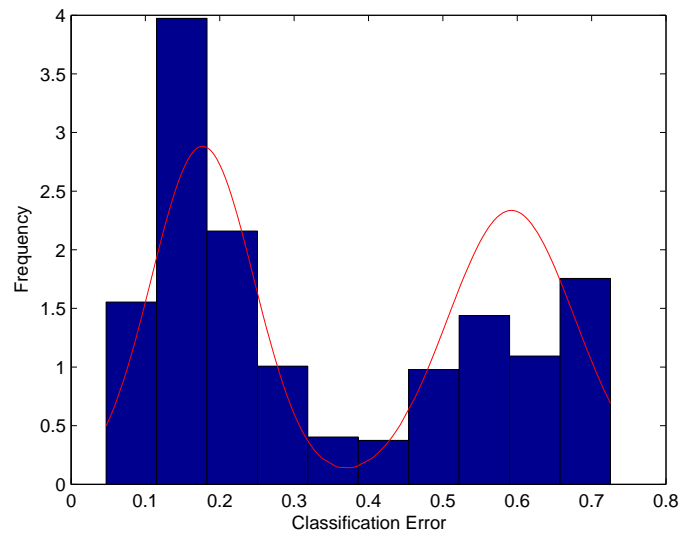


Figure 4.4: Distribution of Response Values from Second GA Experiment.

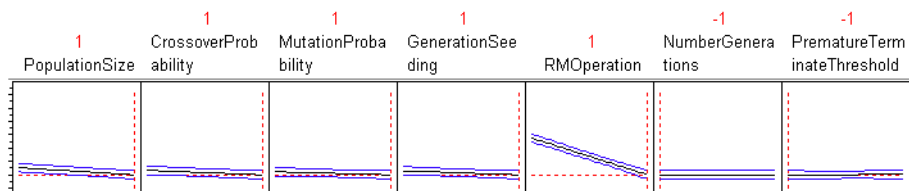


Figure 4.5: Factor Profiler from Second GA Experiment.

# The Symmetric Elliptically-Contoured Mixture Model (ECMM)

“*[Symmetric] book stacking, just like the Philadelphia mass turbulence of 1945.*” -

Ray Stantz, Ghostbusters

## 5.1 Multivariate Symmetric Elliptically-Contoured Distributions

From Anderson and Fang (1990), if the characteristic function of a random vector  $x \in \mathbb{R}^{1 \times p}$  is of the form

$$e^{it\mu} \phi(t'\Sigma t), \quad (5.1)$$

$\mu$  and  $\Sigma$  are of size  $1 \times p$  and  $p \times p$ , respectively,  $\Sigma$  is symmetric and positive definite, and  $\phi$  is a proper scalar function, we say  $x$  is drawn from a multivariate elliptically-contoured distribution (EC). The probability density function for  $x$  is

$$x \sim EC_p(\mu, \Sigma, \phi) = C_p |\Sigma|^{-\frac{1}{2}} g(t_i \Sigma^{-1} t_i'), t_i = (x_i - \mu), \quad (5.2)$$

where  $g$  is the *density generating function*, and  $C_p$  is a normalizing constant that is specific to  $g$ . From Fang et al. (1990), we can compute  $C_p$  by

$$C_p = \frac{\Gamma\left(\frac{p}{2}\right)}{2\pi^{\frac{p}{2}} \int_0^\infty r^{p-1} g(r^2) dr}. \quad (5.3)$$

Three major subclasses of EC distributions for which we go into more detail are:

Pearson Type II:  $f(x)_{PII} = \frac{\Gamma(\frac{p}{2} + \nu + 1)}{\pi^{\frac{p}{2}} \Gamma(\nu + 1)} |\Sigma|^{-\frac{1}{2}} (1 - t\Sigma^{-1}t')^\nu$

Pearson Type VII:  $f(x)_{PVII} = \frac{\Gamma(N)}{(\pi\nu)^{\frac{p}{2}} \Gamma(N - \frac{p}{2})} |\Sigma|^{-\frac{1}{2}} \left(1 + \frac{t\Sigma^{-1}t'}{\nu}\right)^{-N}$

Kotz's Type:  $f(x)_{KT} = \frac{\beta\Gamma(\frac{p}{2})}{\pi^{\frac{p}{2}} \Gamma(\frac{2N+p-2}{2\beta})} r^{\frac{2N+p-2}{2\beta}} |\Sigma|^{-\frac{1}{2}} (t\Sigma^{-1}t')^{N-1} e^{-r(t\Sigma^{-1}t')^\beta}$

As will soon be seen, we can model data with a variety of symmetric shapes using these densities.

## 5.2 Parameter Estimation & Inference

### 5.2.1 Parameter Estimation

For the EC class of distributions, the maximum likelihood estimator for the centroid  $\mu$  is

$$\hat{\mu} = \overline{X} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (5.4)$$

We can determine the maximum likelihood estimators for  $\Sigma$  by the procedure outlined in Anderson and Fang (1990), but we need to assume the density generator  $g(\cdot)$  is a decreasing and differentiable function.

Consider maximizing the function

$$h(\lambda) = \lambda^{-\frac{np}{2}} g\left(\frac{p}{\lambda}\right), \quad (5.5)$$

for positive values of  $\lambda$ . The value  $\lambda_{\max}$  that satisfies the requirements

$$\frac{\partial}{\partial \lambda} h(\lambda_{\max}) = 0, \text{ and } \frac{\partial^2}{\partial \lambda^2} h(\lambda_{\max}) < 0, \quad (5.6)$$

maximizes the likelihood, and gives us the MLE

$$\hat{\Sigma}_{EC} = \lambda_{\max} W, \quad (5.7)$$

where  $W$  is the  $p \times p$  sum of squared errors matrix. The maximized log-likelihood for  $n$  observations, adopted from Anderson and Fang (1990) is shown in (5.8) without proof.

$$l(\hat{\theta} \mid X) = n \log C_p - \frac{np}{2} \log \lambda_{\max} - \frac{n}{2} \log |W| + \log g\left(\frac{p}{\lambda_{\max}}\right) \quad (5.8)$$

While they did not include the  $C_p$  term, we have to, since we need the maximized log-likelihood for model selection using the information criteria. Without it, the comparison would be unfair.

We now have the parameters of the specific pdf generators to estimate. This is a tricky problem requiring (thus far) intractable matrix calculus. One approach is to perform a numerical search. Farrell and Mersereau (2004) developed the EM algorithm for the multivariate Student's T distribution in which they iteratively estimated  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  ( $k = 1, \dots, \hat{K}$ ) for a T mixture model. At each iteration, the bisection method was used to compute MLEs for each of the  $\nu_k$ . This “brute force” method was justified by the high probability that the early EM algorithm estimates were incorrect. Of course, this is a fairly simple case of the EC class. We know that the degrees of freedom must be positive, and that the T distribution approximates the Gaussian as the shape parameter gets large (say,  $\nu > 40$ ). Thus, the initial range could be set to  $[3, 40]$ ; it would only take 12 iterations to bracket the maximum within  $\pm 0.01$ .

However, when fitting the Pearson Type VII distribution, we actually have to estimate both  $N$  and  $\nu$ . For the Kotz subclass, we need  $N$ ,  $r$ , and  $\beta$ . Thus, a numerical bracketing search would have to partition a 2-dimensional and 3-dimensional surface, respectively. This would substantially increase the computational burden required to estimate the pdf generator-specific parameters. Like Farrell and Mersereau (2004), our response is to fix some parameters and let others be estimated.

For the Pearson Type VII distribution, we set  $N = (p + \nu) / 2$  and estimate the degrees of freedom  $\nu$  for the multivariate T using a method of moments procedure. Sutradhar and Ali (1986) considered multivariate regression under the assumption that the error terms,  $\varepsilon_i$ , were drawn from a multivariate T distribution. Using the 4<sup>th</sup> moment (related to kurtosis), they found

$$\hat{\nu} = 2 \frac{3 \sum_{i=1}^p (\hat{\sigma}_{ii}^2)^2 - \frac{2}{n} \sum_{i=1}^p \sum_{j=1}^n \varepsilon_{ij}^4}{3 \sum_{i=1}^p (\hat{\sigma}_{ii}^2)^2 - \frac{1}{n} \sum_{i=1}^p \sum_{j=1}^n \varepsilon_{ij}^4}, \quad (5.9)$$

where  $\hat{\sigma}_{ii}^2$  indicates the variance in the  $i^{\text{th}}$  dimension. Now, rather than error terms, we have the observations  $x_i$  drawn from a Pearson Type VII distribution with the assumption that  $N =$



$(p + \nu)/2$ . In partially vectorized notation, we rewrite (5.9) as

$$\hat{\nu} = 2 \frac{\frac{3}{n^2} \sum_{i=1}^p w_{ii}^2 - \frac{2}{n} \sum_{i=1}^{np} \text{Vec}(x)_i^4}{\frac{3}{n^2} \sum_{i=1}^p w_{ii}^2 - \frac{1}{n} \sum_{i=1}^{np} \text{Vec}(x)_i^4}. \quad (5.10)$$

We use  $w_{ii}$  to indicate the  $i^{\text{th}}$  diagonal element of  $W$ .  $\text{Vec}(\cdot)$  is the vector operator, that takes the columns of a matrix and vertically catenates them. For example:

$$\text{Vec} \left( \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \right) \longrightarrow \begin{bmatrix} 1 \\ 3 \\ 2 \\ 4 \end{bmatrix}.$$

Clearly, this estimator can return an invalid negative value. When this happens, we resort to a numerical search, like Farrell and Mersereau (2004). We maximize the profile likelihood of the PVII distribution w.r.t.  $\nu$  in the interval  $[3, 100]$ . The actual function we maximize is

$$\log L(\nu \mid X) = n \log \frac{\Gamma\left(\frac{p+\nu}{2}\right)}{(\pi\nu)^{\frac{p}{2}} \Gamma\left(\frac{\nu}{2}\right)} - \frac{n}{2} \log |\hat{\Sigma}_{PVII}| - \frac{p+\nu}{2} \sum_{i=1}^n \log \left( 1 + \frac{t_i \hat{\Sigma}_{PVII} t_i'}{\nu} \right). \quad (5.11)$$

There seems to be little point in searching beyond  $\hat{\nu} = 100$ , since at this point the PVII distribution approximates the Gaussian very closely.

For Kotz's type, we set  $r = \frac{1}{2}$  and  $N = 1$  - thus it is reduced to the multivariate power exponential distribution. Several methods have been proposed for estimating the shape parameter  $\beta$  of the PE distribution. If we define

$$d_i^2 = \left[ (x_i - \hat{\mu})' \hat{\Sigma}^{-1} (x_i - \hat{\mu})' \right]^2, \quad (5.12)$$

where  $\overline{d^2}$  and  $\sigma_d^2$  are the mean and variance of the squared Mahalanobis distances, we have

$$\text{Bozdogan (1995): } \hat{\beta} = \frac{p}{4\overline{d^2}}$$

$$\text{Bozdogan (1995): } \hat{\beta} = \frac{\overline{d^2}}{\sigma_d^2}$$

$$\text{using results of Seo and Toyama (1996): } \hat{\beta} = \frac{\overline{d^2}}{p(p+2)} - 1 \quad (p+2) - 1$$

Finally, based on MoM estimators, we can get  $\hat{\beta}$  as the root of the equations

$$\frac{2^{\frac{1}{\beta}} \Gamma\left(\frac{p+2}{2\beta}\right)}{\Gamma\left(\frac{p}{2\beta}\right)} - \overline{d^2} = 0, \quad (5.13)$$

or

$$\frac{p^2 \Gamma\left(\frac{p}{2\beta}\right) \Gamma\left(\frac{p+4}{2\beta}\right)}{\Gamma^2\left(\frac{p+2}{2\beta}\right)} - \overline{d^2} = 0. \quad (5.14)$$

For these estimators, see Bozdogan (1995) and Liu (2006), respectively. In simulation studies, we have found the last method consistently produces more accurate estimates of the true shape parameter than the others - our numerical results use this method.

Before going on, we feel the need to provide some justification for reducing the EC class of distributions to just the T or PE. On the surface, this may seem overly restrictive. However, flipping ahead to Figures 5.3 through 5.6, we see that these two distributions allow us to fit both spherically- and elliptically-contoured data. Additionally, between the Pearson Type VII and Kotz type subclasses, we can simultaneously or separably model both tail and peak behavior.

### 5.2.2 Inference

If we want to compute interval estimates, rather than just point estimates for  $\mu$  or  $\Sigma_{EC}$ , we need the Fisher information matrix. Of course, this is also required for *ICOMP*. Here we make use of the results of Liu (2002), in which the Fisher information matrix was derived in the context of multivariate regression with EC error terms. First, however, we need some preliminary definitions.

Let

$$G(t) = \frac{\partial}{\partial t} \log g(t) = \frac{g'(t)}{g(t)}, \quad (5.15)$$

and

$$J(t) = \frac{\partial}{\partial t} G(t). \quad (5.16)$$

The observed inverse information matrix is then

$$\hat{\mathcal{F}}^{-1} = \begin{bmatrix} -\frac{1}{2n\hat{G}} \hat{\Sigma}_{EC} & \mathbf{0} \\ \mathbf{0} & H^{-1} \end{bmatrix}, \quad (5.17)$$

where  $\hat{G} = G\left(\frac{p}{\lambda_{\max}}\right)$ . Using the general result that  $\hat{G} = -n\lambda_{\max}/2$ , the 4<sup>th</sup> orthant is

$$H = D'_p \left( \frac{n}{2} \hat{\Sigma}_{EC}^{-1} \otimes \hat{\Sigma}_{EC}^{-1} - \frac{\hat{J}}{\lambda_{\max}^2} \text{Vec}(\hat{\Sigma}_{EC}^{-1}) \text{Vec}'(\hat{\Sigma}_{EC}^{-1}) \right) D_p, \quad (5.18)$$

we define  $\hat{J}$  to be  $J\left(\frac{p}{\lambda_{\max}}\right)$ .

The published research regarding estimation of the model covariance matrix for the EC distribution seems to be limited to the regression context. As such, we have had to generalize the Liu's results. Recall that the multivariate regression model can be expressed in two ways:

$$\underset{n \times p}{Y} = \underset{n \times q}{X} \underset{q \times p}{B} + \underset{n \times p}{E} \quad \text{or} \quad Y_i \sim \text{Dist}(X_i B \mid \theta).$$

Here, we use  $\sim \text{Dist}(\cdot)$  to indicate some arbitrary distribution. Generalizing this to the situation where the values of  $Y$  are identically distributed, we have

$$Y \sim \text{Dist}(XB^* \mid \theta), \quad (5.19)$$

where  $X$  is a matrix of 1's. Hence, we see that for a sample of size  $n$  and  $q = 1$ ,  $X'X = n$ . This justifies the way we replaced Liu's  $(X'X)^{-1}$  with  $1/n$  in the second orthant of (5.17). Finally, we would note that we compute the FIM solely based on  $\mu$  and  $\Sigma_{EC}$  - the shape parameters that we estimate are not counted as parameters. Hence, any measure of EC model complexity based on this simplified FIM will likely underestimate estimation uncertainty and complexity.

## 5.3 Details for EC Subclasses

Here we take the general results presented thus far, and derive details for Pearson Type II, Pearson Type VII, and Kotz's Type distributions. Some estimation details shown here were taken from Liu (2006).

### 5.3.1 Pearson Type II

When the probability density generating function looks like

$$g(u) = (1-u)^\nu, \nu > -1, \quad (5.20)$$

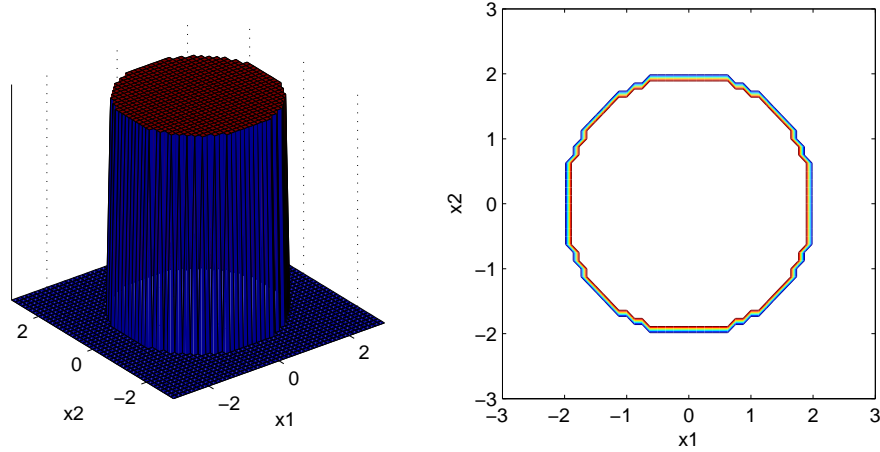


Figure 5.1: Pearson Type II Subclass for  $\nu = 0$ .

we say  $X$  follows a symmetric Pearson Type II distribution. The normalizing constant is shown in (5.21).

$$C_p = \frac{\Gamma\left(\frac{p}{2} + \nu + 1\right)}{\pi^{\frac{p}{2}} \Gamma(\nu + 1)} \quad (5.21)$$

If  $X \sim PII_p(\mu, \Sigma, \nu)$ , we have the pdf

$$f(x)_{PII} = \frac{\Gamma\left(\frac{p}{2} + \nu + 1\right)}{\pi^{\frac{p}{2}} \Gamma(\nu + 1)} |\Sigma|^{-\frac{1}{2}} (1 - t\Sigma^{-1}t')^\nu. \quad (5.22)$$

For this to be a true density, we require that  $0 \leq (t\Sigma^{-1}t') \leq 1$ , which restricts its usefulness somewhat. In fact, though we show it here for completeness, we won't use it in our numerical results. Figures 5.1 and 5.2 show the density surface and contours for the Pearson Type II distribution with  $\mu = \mathbf{0}$  and  $\Sigma = I_p$  and  $\nu = 0, 1$ .

For the Pearson Type II subclass, the solution for (5.6) is

$$\lambda_{\max} = \frac{2\nu + np}{n}, \quad (5.23)$$

so the MLE for the Pearson Type II covariance matrix is shown in (5.24).

$$\hat{\Sigma}_{PII} = \frac{2\nu + np}{n} W \quad (5.24)$$

For computing the estimated model variance-covariance matrix, we need  $\hat{G}$  and  $\hat{J}$ . From (5.15)

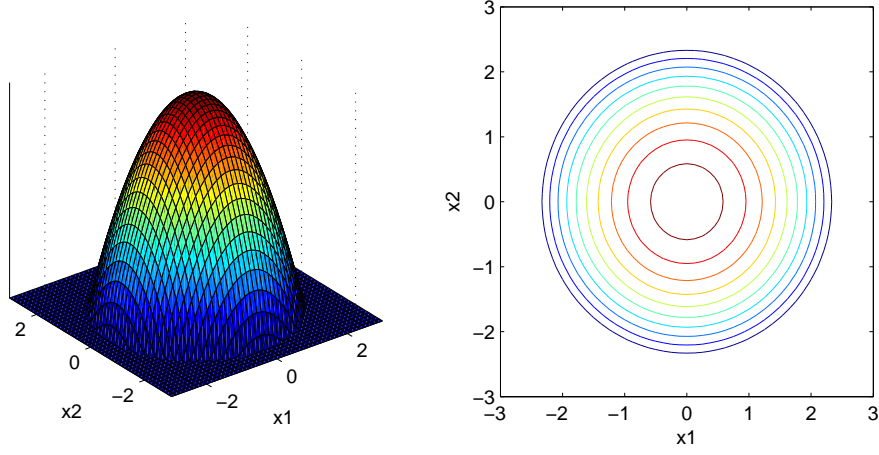


Figure 5.2: Pearson Type II Subclass for  $\nu = 1$ .

and (5.16), we derive

$$G = \frac{\partial}{\partial t} \log g(t) = \frac{g'(t)}{g(t)} = -\frac{\nu}{1-t} \longrightarrow \hat{G} = -\frac{2\nu + np}{2}, \quad (5.25)$$

and

$$J(t) = \frac{\partial}{\partial t} G(t) = -\frac{\nu}{(1-t)^2} \longrightarrow \hat{J} = -\frac{(2\nu + np)^2}{4\nu}. \quad (5.26)$$

The estimated IFIM is then

$$\hat{\mathcal{F}}_{PII}^{-1} = \begin{bmatrix} \frac{1}{2\nu+np} \frac{1}{n} \hat{\Sigma}_{PII} & \mathbf{0} \\ \mathbf{0} & H^{-1} \end{bmatrix}. \quad (5.27)$$

For computing  $H$ , we see that

$$\frac{\hat{J}}{\lambda_{\max}^2} = \frac{-\frac{(2\nu+np)^2}{4\nu}}{\frac{(2\nu+np)^2}{n^2}} = -\frac{n^2}{4\nu}.$$

Therefore,  $H$  is

$$H = D'_p \left( \frac{n}{2} \hat{\Sigma}_{PII}^{-1} \otimes \hat{\Sigma}_{PII}^{-1} + \frac{n^2}{4\nu} \text{Vec}(\hat{\Sigma}_{PII}^{-1}) \text{Vec}'(\hat{\Sigma}_{PII}^{-1}) \right) D_p. \quad (5.28)$$

Finally, the maximized log-likelihood is

$$\log L(\hat{\theta} \mid X) = n \log \frac{\Gamma\left(\frac{p}{2} + \nu + 1\right)}{\pi^{\frac{p}{2}} \Gamma(\nu + 1)} - \frac{np}{2} \log \frac{2\nu + np}{n} - \frac{n}{2} \log |W| + \nu \log \frac{2\nu}{2\nu + np}. \quad (5.29)$$

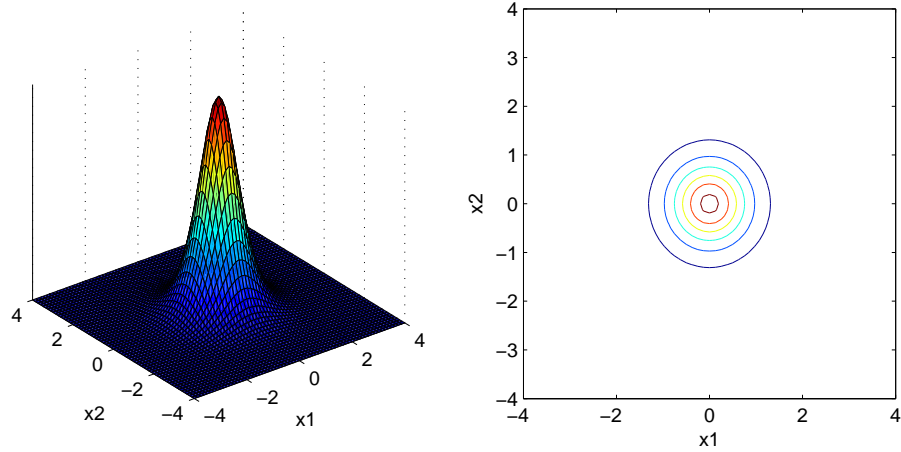


Figure 5.3: Pearson Type VII Subclass for  $\nu = 3$ .

### 5.3.2 Pearson Type VII

Next, we have the Pearson Type VII distribution with pdf generator and normalizing constant shown in (5.30) and (5.31).

$$g(u) = \left(1 + \frac{u}{\nu}\right)^{-N}, N > \frac{p}{2}, \nu > 0 \quad (5.30)$$

$$C_p = \frac{\Gamma(N)}{(\pi\nu)^{\frac{p}{2}} \Gamma(N - \frac{p}{2})} \quad (5.31)$$

When  $N = (p + \nu)/2$  and  $\nu > 2$ , we recognize the multivariate Student's T distribution, as previously mentioned; if  $\nu = 1$  and  $N = (p + 1)/2$ , the Pearson Type VII reduces to the multivariate Cauchy distribution. If we have  $X \sim PVII_p(\mu, \Sigma, \nu, N)$ , the pdf is given by

$$f(x)_{PVII} = \frac{\Gamma(N)}{(\pi\nu)^{\frac{p}{2}} \Gamma(N - \frac{p}{2})} |\Sigma|^{-\frac{1}{2}} \left(1 + \frac{t\Sigma^{-1}t'}{\nu}\right)^{-N}. \quad (5.32)$$

Figure 5.3 plots the density surface and contours for the PVII distribution using  $N = 1$ ,  $\nu = 3$ ,  $\mu = \mathbf{0}$ , and  $\Sigma = I_p$ . Figure 5.4 shows the same plots for  $\nu = 20$ .

The maximum likelihood estimator for the Pearson Type VII covariance matrix is

$$\hat{\Sigma}_{PVII} = \lambda_{\max} W = \frac{2N - p}{n\nu} W. \quad (5.33)$$

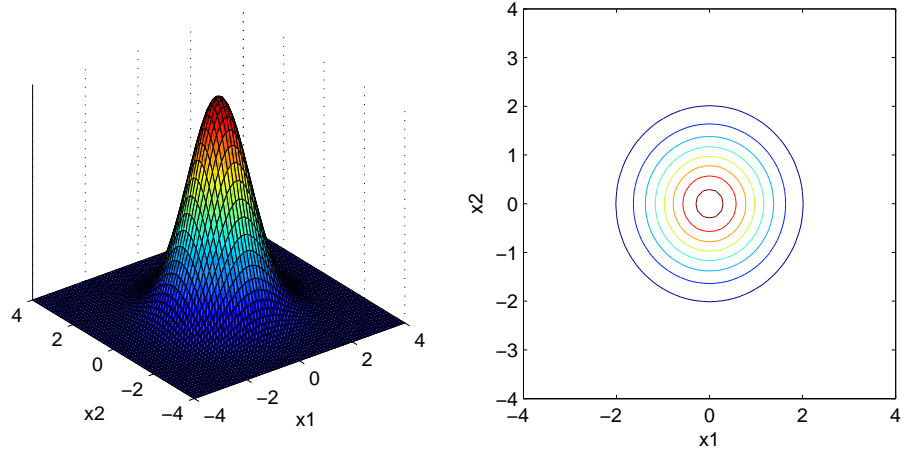


Figure 5.4: Pearson Type VII Subclass for  $\nu = 20$ .

For computing the IFIM, we have

$$G(t) = -\frac{N}{\nu} \frac{(1 + \frac{t}{\nu})^{-N-1}}{(1 + \frac{t}{\nu})^{-N}} = -\frac{N}{\nu + t} \longrightarrow \hat{G} = G\left(\frac{p}{\lambda_{\max}}\right) = -\frac{N(2N-p)^2}{(2N-p+np)\nu}, \quad (5.34)$$

and

$$J(t) = \frac{N}{(\nu + t)^2} \longrightarrow \hat{J} = \frac{N(2N-p)^2}{(2N-p+np)\nu^2}. \quad (5.35)$$

We also have

$$H = D'_p \left( \frac{n}{2} \hat{\Sigma}_{PVII}^{-1} \otimes \hat{\Sigma}_{PVII}^{-1} - \frac{Nn^2}{(2N-p+np)^2} \text{Vec}(\hat{\Sigma}_{PVII}^{-1}) \text{Vec}'(\hat{\Sigma}_{PVII}^{-1}) \right) D_p, \quad (5.36)$$

since  $\frac{\hat{J}}{\lambda_{\max}^2} = (Nn^2) / (2N-p+np)^2$ . The estimated IFIM is shown in (5.37)

$$\hat{\mathcal{F}}_{PVII}^{-1} = \begin{bmatrix} \frac{\nu}{2N-np} \frac{1}{n} \hat{\Sigma}_{PVII} & \mathbf{0} \\ \mathbf{0} & H^{-1} \end{bmatrix}. \quad (5.37)$$

We also have the maximized likelihood

$$\log L(\hat{\theta} \mid X) = n \log \frac{\Gamma(N)}{(\pi\nu)^{\frac{p}{2}} \Gamma(N - \frac{p}{2})} - \frac{np}{2} \log \frac{2N-p}{n\nu} - \frac{n}{2} \log |W| - N \log \frac{2N}{2N-p}. \quad (5.38)$$

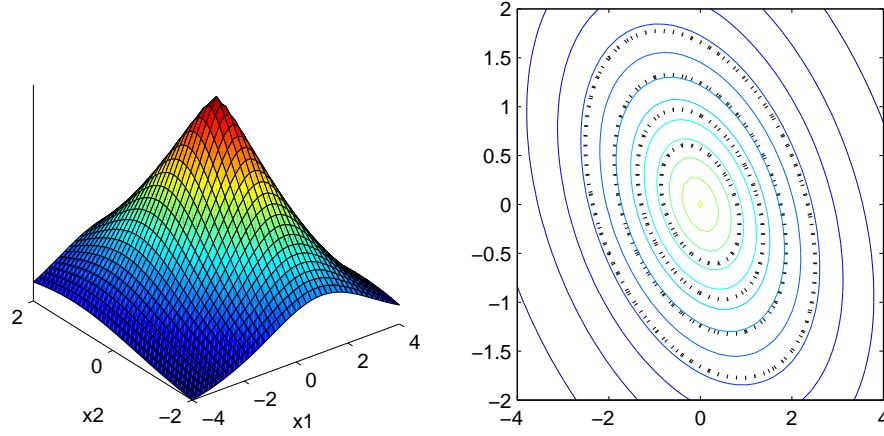


Figure 5.5: Kotz Subclass Reduced to PE Reduced to the Laplace.

### 5.3.3 Kotz's Type

If  $X \sim EC_p(\mu, \Sigma, \phi)$ , and the pdf generator is of the form

$$g(u) = u^{N-1} e^{-ru^\beta}, \quad (5.39)$$

we say  $X$  is drawn from a symmetric Kotz type distribution. We require  $r, \beta > 0$ , and  $2N + p > 2$ . The normalizing constant for this subclass, from (5.3), is shown in (5.40).

$$C_p = \frac{\beta \Gamma\left(\frac{p}{2}\right)}{\pi^{\frac{p}{2}} \Gamma\left(\frac{2N+p-2}{2\beta}\right)} r^{\frac{2N+p-2}{2\beta}} \quad (5.40)$$

Thus, the entire pdf is

$$f(x)_{KT} = \frac{\beta \Gamma\left(\frac{p}{2}\right)}{\pi^{\frac{p}{2}} \Gamma\left(\frac{2N+p-2}{2\beta}\right)} r^{\frac{2N+p-2}{2\beta}} |\Sigma|^{-\frac{1}{2}} (t\Sigma^{-1}t')^{N-1} e^{-r(t\Sigma^{-1}t')^\beta}. \quad (5.41)$$

If we set  $N = 1$  and  $r = \frac{1}{2}$ , this reduces to the PE distribution. If we further set  $\beta = 1$ , we get the multivariate Gaussian;  $\beta = \frac{1}{2}$  gives us the multivariate Laplace distribution. As  $\beta \rightarrow \infty$ , the Kotz distribution (for  $N = 1$  and  $r = \frac{1}{2}$ ) approximates the multivariate uniform distribution. In Figures 5.5 and 5.6, we have the surface and contour plots for two special cases of the Kotz distribution. The black dotted contours are for the multivariate Gaussian distribution with the same mean and covariance matrix.



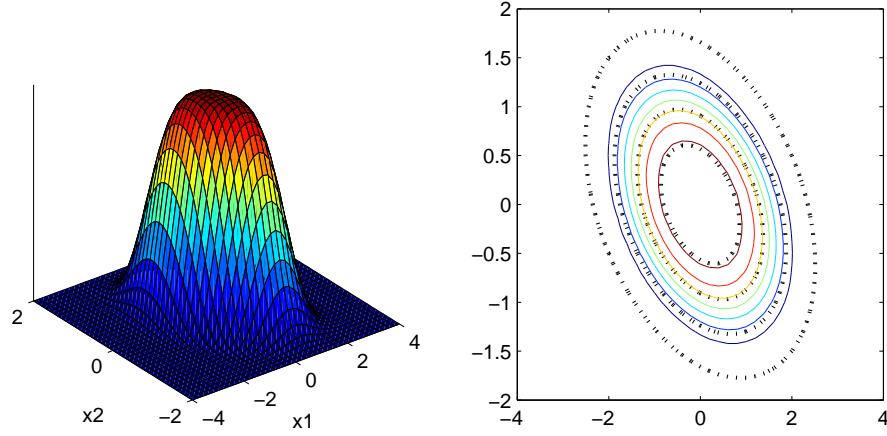


Figure 5.6: Kotz Subclass Reduced to PE Approximating the Uniform.

Using the pdf generator in (5.39), we have

$$h(\lambda) = \lambda^{-\frac{np}{2}} \left(\frac{p}{\lambda}\right)^{N-1} e^{-r\left(\frac{p}{\lambda}\right)^\beta} \quad (5.42)$$

from (5.5). The multiplier for estimating  $\Sigma_{KT}$ ,  $\lambda_{\max}$ , solves

$$\frac{\partial^2}{\partial \lambda^2} h(\lambda) = \lambda^{-N-\frac{np}{2}} e^{-r\left(\frac{p}{\lambda}\right)^\beta} \left( r s \left(\frac{p}{\lambda}\right)^\beta - N - \frac{np}{2} + 1 \right) = 0. \quad (5.43)$$

The root for this is shown in (5.44).

$$\lambda_{\max} = p \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{-\frac{1}{\beta}} \quad (5.44)$$

Thus, the EC covariance matrix estimate is

$$\hat{\Sigma}_{KT} = p \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{-\frac{1}{\beta}} W. \quad (5.45)$$

For the inverse observed information matrix, we have

$$G(t) = \frac{N - 1 - r\beta t^2}{t} \longrightarrow \hat{G} = -\frac{np}{2} \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{-\frac{1}{\beta}}, \text{ and}$$

$$J(t) = \frac{1 - N - r(\beta^2 - \beta)t^\beta}{t^2} \longrightarrow \hat{J} = \left[ 1 - N - (\beta - 1) \left( N + \frac{np}{2} - 1 \right) \right] \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{-\frac{2}{\beta}}.$$

The model covariance matrix for the Kotz subclass of distributions is then shown in (5.46)

$$\hat{\mathcal{F}}_{KT}^{-1} = \begin{bmatrix} \frac{1}{np} \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{\frac{1}{\beta}} \frac{1}{n} \hat{\Sigma}_{KT} & \mathbf{0} \\ \mathbf{0} & H^{-1} \end{bmatrix}, \quad (5.46)$$

with  $H$  defined as shown here.

$$H = D_p' \left( \frac{n}{2} \hat{\Sigma}_{KT}^{-1} \otimes \hat{\Sigma}_{KT}^{-1} - \frac{[1 - N - (\beta - 1)(N + \frac{np}{2} - 1)]}{p^2} \text{Vec}(\hat{\Sigma}_{KT}^{-1}) \text{Vec}'(\hat{\Sigma}_{KT}^{-1}) \right) D_p. \quad (5.47)$$

Finally, we have the maximized log likelihood for this EC subclass from (5.8):

$$\begin{aligned} \log L(\hat{\theta} \mid X) &= n \log \frac{\beta \Gamma(\frac{p}{2})}{\pi^{\frac{p}{2}} \Gamma(\frac{2N+p-2}{2\beta})} + \frac{2N+p-2}{2\beta} \log r - \frac{np}{2} \log \left[ p \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{-\frac{1}{\beta}} \right] - \dots \\ &\quad - \frac{n}{2} \log |W| + \frac{N-1}{\beta} \log \frac{N + \frac{np}{2} - 1}{r\beta} - \frac{N + \frac{np}{2} - 1}{\beta} \end{aligned} \quad (5.48)$$

For the special case of the Gaussian distribution, if we fill in the values  $N = \beta = 1$  and  $r = \frac{1}{2}$ , we have the following quantities.

$$\hat{\Sigma} = \lambda_{\max} W = p \left( \frac{N + \frac{np}{2} - 1}{r\beta} \right)^{-\frac{1}{\beta}} W = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})' (x_i - \bar{X}) \quad (5.49)$$

$$\hat{\mathcal{F}}^{-1} = \begin{bmatrix} \frac{1}{n} \hat{\Sigma} & \mathbf{0} \\ \mathbf{0} & \frac{2}{n} D_p^+ (\hat{\Sigma} \otimes \hat{\Sigma}) D_p^{+'} \end{bmatrix} \quad (5.50)$$

$$\log L(\hat{\theta} \mid X) = \frac{np}{2} \log(2\pi) - \frac{n}{2} \log |\hat{\Sigma}| - \frac{np}{2} \quad (5.51)$$

It is easily verified that (5.49) through (5.51) are the correct quantities for the multivariate Gaussian distribution, as expected.

## 5.4 Hybrid EM Algorithm for the ECMM

Here we extend the EM algorithm from the GMM case to include other special cases of the elliptically-contoured distributions by using method-of-moments and/or numerical search to estimate the shape parameter. Hence we call it a ‘‘Hybrid EM’’. The **E**-step is conceptually no different than already shown for the GMM. At the  $t^{\text{th}}$  iteration, the algorithm estimates the pos-

terior probabilities of group membership for datapoint  $i$  and mixture  $k$  using (5.52).

$$\hat{p}_i(k) = \frac{\hat{\pi}_k^{(t-1)} f_{ECk}(x_i | \hat{\theta}_k^{(t-1)})}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k^{(t-1)} f_{ECk}(x_i | \hat{\theta}_k^{(t-1)})} \quad (5.52)$$

For the **M**-step, we extend the results of Farrell and Mersereau (2004) to re-estimate  $\pi_k$ ,  $\mu_k$ , and  $\Sigma_k$  for  $k = 1, 2, \dots, \hat{K}$ . Their EM implementation for these parameters was identical to that of the Gaussian mixture model:

$$\hat{\pi}_k^{(t)} = \frac{1}{n} \sum_{i=1}^n \hat{p}_i(k), \quad (5.53)$$

$$\hat{\mu}_k^{(t)} = \frac{1}{n \hat{\pi}_k^{(t)}} \sum_{i=1}^n x_i \hat{p}_i(k), \quad (5.54)$$

$$\hat{\Sigma}_k^{(t)} = \frac{1}{n \hat{\pi}_k^{(t)}} \sum_{i=1}^n \hat{p}_i(k) (x_i - \hat{\mu}_k^{(t)})' (x_i - \hat{\mu}_k^{(t)}). \quad (5.55)$$

The problem here is with  $\hat{\Sigma}_i^{(t)}$  - as already shown in this chapter, (5.55) won't generally maximize the likelihood for the EC class of distributions (except in the special case of the Kotz with  $N = \beta = 1, r = \frac{1}{2}$ ). Thus, we can define

$$W_k^{(t)} = \sum_{i=1}^n \hat{p}_i(k) (x_i - \hat{\mu}_k^{(t)})' (x_i - \hat{\mu}_k^{(t)}), \quad (5.56)$$

and estimate the covariance matrix in the  $t^{\text{th}}$  iteration as

$$\hat{\Sigma}_{PVIIk}^{(t)} = \frac{p + \nu_k^{(t)} - p}{\nu_k^{(t)} n \hat{\pi}_k^{(t)}} W_k^{(t)}, \text{ or} \quad (5.57)$$

$$\hat{\Sigma}_{KTK}^{(t)} = p \left( \frac{n \hat{\pi}_k^{(t)} p}{\beta_k^{(t)}} \right)^{-\frac{1}{\beta_k^{(t)}}} W_k^{(t)}, \quad (5.58)$$

rather than (5.55).

If we are fitting mixtures of the Pearson Type VII subclass, at each iteration of the EM algorithm, we modify the already shown MoM estimator due to Sutradhar and Ali (1986) to compute

$\nu_k^{(t)}$ , as shown in (5.59).

$$\hat{\nu}_k^{(t)} = 2 \frac{\frac{3}{n^{2*}} \sum_{i=1}^p (w_{kii}^{(t)*})^2 - \frac{2}{n^*} \sum_{i=1}^{n^*p} (Vec(x)_i^*)^4}{\frac{3}{n^{2*}} \sum_{i=1}^p (w_{kii}^{(t)*})^2 - \frac{1}{n^*} \sum_{i=1}^{n^*p} (Vec(x)_i^*)^4} \quad (5.59)$$

Here,  $w_{kii}^{(t)}$  indicates the  $i^{\text{th}}$  diagonal entry of  $W_k^{(t)}$ , and the  $*$  indicates that the MAP rule is used to identify datapoints most likely to belong to the current group. For cases when the MoM estimator is negative, we adopt and maximize (5.11) for the hybrid EM algorithm, using the MAP rule.

$$\begin{aligned} \log L(\nu_k^{(t)} | X_k) &= n\hat{\pi}_k^{(t)} \log \frac{\Gamma\left(\frac{p+\nu_k^{(t)}}{2}\right)}{(\hat{\pi}_k^{(t)} \nu_k^{(t)})^{\frac{p}{2}} \Gamma\left(\frac{\nu_k^{(t)}}{2}\right)} - \frac{n\hat{\pi}_k^{(t)}}{2} \log |\hat{\Sigma}_{PVIIk}^{(t)}| \\ &\quad - \frac{p + \nu_k^{(t)}}{2} \log \left(1 + \frac{u_k^{(t)}}{\nu_k^{(t)}}\right). \end{aligned} \quad (5.60)$$

For the Kotz type distribution, we estimate  $\beta_k^{(t)}$  using the MoM estimator of Liu (2006) already shown. In the mixture EM context, we define the Mahalanobis distance as

$$d_i^{2(t)*} = \left[ (x_i - \hat{\mu}_k^{(t)}) S_k^{2(t)*} (x_i - \hat{\mu}_k^{(t)})' \right]^2, \quad (5.61)$$

where  $S_k^{2(t)*}$  is computed based on the MAP rule. For  $\overline{d^{2(t)*}}$  in the following equation, the MAP rule is also used, so the Mahalanobis distance is only computed for datapoints that are likely to be in the  $k^{\text{th}}$  group. Rather than finding the root of (5.14) directly, we use a potentially more stable implementation. Since  $\log(\cdot)$  is a monotonic function, finding a root of

$$\frac{p^2 \Gamma\left(\frac{p}{2\hat{\beta}_k^{(t)}}\right) \Gamma\left(\frac{p+4}{2\hat{\beta}_k^{(t)}}\right)}{\Gamma^2\left(\frac{p+2}{2\hat{\beta}_k^{(t)}}\right)} - \overline{d^{2(t)*}} = 0$$

is the same as minimizing

$$\log \frac{p^2 \Gamma\left(\frac{p}{2\hat{\beta}_k^{(t)}}\right) \Gamma\left(\frac{p+4}{2\hat{\beta}_k^{(t)}}\right)}{\Gamma^2\left(\frac{p+2}{2\hat{\beta}_k^{(t)}}\right)} - \log \overline{d^{2(t)*}}. \quad (5.62)$$

We set practical limits on the Kotz shape parameter  $\beta_k^{(t)} = [0.1, 10]$ ; for  $\beta > 10$ , the shape of the distribution does not change much. Thus, we actually use a nonlinear bounded search (golden search) to minimize (5.62).

## 5.5 Information Criteria for the ECMM

When we fit the ECMM with  $\hat{K}$  groups to a  $p$ -dimensional dataset, the number of parameters is as shown here:

$$m = \hat{K}p + \hat{K}\frac{p(p+1)}{2} + \hat{K} + (\hat{K} - 1). \quad (5.63)$$

As with the GMM, we have  $p$  elements of the location vector,  $p(p+1)/2$  unique elements of the scatter matrix, and a single mixing parameter for each cluster. However, we also have a shape parameter ( $\nu$  or  $\beta$ ) to estimate for each group. The ECMM log-likelihood, based on (5.32) and (5.41), is shown here.

$$\log L(\hat{\theta} \mid X) = \sum_{i=1}^n \log \left[ \sum_{k=1}^{\hat{K}} \hat{\pi}_k f_{EC,k}(x_i \mid \hat{\theta}_k) \right] \quad (5.64)$$

Thus, we have *AIC* and *SBC* for the ECMM in (5.65) and (5.66) - of course, these don't look any different than previously shown.

$$AIC = -2 \log L(\hat{\theta} \mid X) + 3m \quad (5.65)$$

$$SBC = -2 \log L(\hat{\theta} \mid X) + \log(n) m \quad (5.66)$$

For computing *ICOMP*, the inverse Fisher information matrix is of the same form as for the GMM model:

$$\hat{\mathcal{F}}^{-1} = \begin{bmatrix} \hat{\mathcal{F}}_{\pi}^{-1} & & & \mathbf{0} \\ & \hat{\mathcal{F}}_1^{-1} & & \\ & & \ddots & \\ \mathbf{0} & & & \hat{\mathcal{F}}_{\hat{K}}^{-1} \end{bmatrix}. \quad (5.67)$$

The first block is no different than what was already shown in (3.32). Block  $k+1$  is the IFIM computed from the  $k^{\text{th}}$  group, based on the specific EC subclass used ((5.37) or (5.46)). We then

have  $ICOMP$  and  $ICOMP_{PEU}$  for the EC mixture model:

$$ICOMP(\hat{\mathcal{F}}^{-1}) = -2 \log L(\hat{\theta} \mid X) + 2C_1(\hat{\mathcal{F}}^{-1}), \quad (5.68)$$

$$ICOMP_{PEU}(\hat{\mathcal{F}}^{-1}) = -2 \log L(\hat{\theta} \mid X) + m + \log(n) C_1(\hat{\mathcal{F}}^{-1}). \quad (5.69)$$

In both cases, we save a potentially significant amount of computational resources (time & storage) by not actually building the individual matrixes. Recall from Chapter 3 that the complexity of  $\hat{\mathcal{F}}^{-1}$  is computed entirely from traces and determinants. For a block diagonal matrix

$$B = \begin{bmatrix} B_1 & & \mathbf{0} \\ & B_2 & \\ & & \ddots \\ \mathbf{0} & & & B_b \end{bmatrix},$$

the trace and determinant are simply

$$tr(B) = \sum_{i=1}^b tr(B_i) \quad \text{and,} \quad |B| = \prod_{i=1}^b |B_i|. \quad (5.70)$$

Thus, when computing either form of  $ICOMP$ , we compute the trace and determinant for  $\hat{\mathcal{F}}_{\pi}^{-1}$  and each mixing distribution, then sum or multiply them as appropriate.

Now we are in a position to determine which of the two EC subclasses might fit a certain dataset the best. As in Section 3.5, we first fit a mixture model of exactly  $K_{\max}$  groups to a given dataset for both the Kotz Type and Pearson Type VII. Whichever model produces the minimum information criterion score is selected for use with that dataset in the subsequent complete analysis.

# The Kernel Density Estimator Mixture Model (KMM)

*“Lorraine, my **density** has popped me to you.”* - George McFLy, Back to the Future

## 6.1 Kernel Density Estimators

For many datasets, enforcing any functional form may not provide a very good fit to the data, leading to suboptimal class assignments. Figure 6.1 is an example of data that does not fit any of the distributions already shown. The right pane shows the density contours, with those for the multivariate Gaussian density in black dots. Clearly, fitting a symmetrical distribution to data from this density will lead to a bad fit. An obvious solution would be to fit a skew-elliptical distribution to data that exhibit asymmetry. Multivariate asymmetrical distributions are a relatively new phenomenon in statistics; an excellent source on this is Ed. M Genton (2004). However, in this chapter, we propose to utilize an even more general nonparametric technique called kernel density estimation (KDE). Though it didn’t see much use for a long time, nonparametric density estimation of this form was first published by Rosenblatt (1956). Kernel density estimation relies on every datapoint to compute the probability density for a given dataset. Hence, we see why kernel methods languished, they require a heavy computational effort. The density estimate for  $x_i$  is computed as a weighted sum that is a function of the distance from  $x_i$  to all other datapoints, such that closer

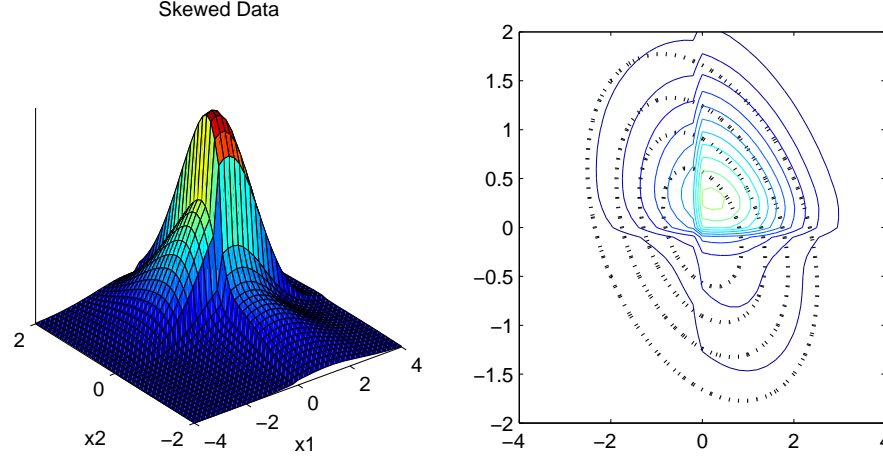


Figure 6.1: Example of Skewed Bivariate Data.

datapoints have a stronger influence on the density.

$$g(x_i) = \frac{1}{nh} \sum_{l=1}^n K\left(\frac{x_l - x_i}{h}\right). \quad (6.1)$$

Figure 6.2 is a good example of how kernel density estimation works - near the mode, the density contributions are more dense than at the tails. The parameter  $h$  (ok, so this is not entirely nonparametric) is called a *bandwidth* or *window width*. We will come back to this in the ensuing section. As long as  $K(t)$  is defined such that it integrates to 1,  $g$  is a proper probability density function. In Table 6.1, we see a list of five common kernel functions, where  $t = (x_l - x_i)/h$ ; their shapes are also shown in shown in Figure 6.3. In the right pane, we see the Gaussian kernel with support of the entire real number line, while the other five have support  $-1 \leq t \leq 1$ . Note that the major difference between these kernel functions is the tail / peak behavior. It would seem that with KDE, selecting the best kernel function is critical; after all, this is the case when fitting a distribution to data. However, as shown by Scott (1992), the actual kernel function used has little effect on the density estimates. The tail and peak variations are smoothed out by the averaging process.

Thus far, we have looked at univariate KDE, but for it to be really useful, we need the ability to compute kernel density estimates for multivariate data. The simplest approach is the product



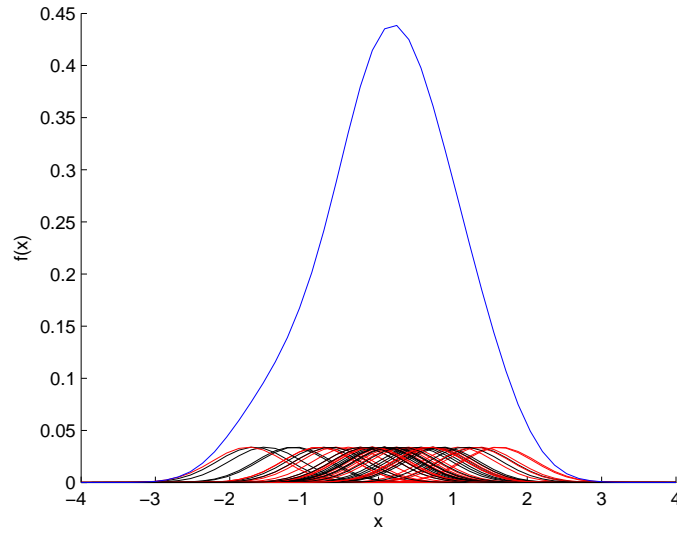


Figure 6.2: Demonstrating Kernel Density Estimate Computation.

Table 6.1: Sample Univariate Kernel Functions.

Kernel	$K(t)$
Biweight	$\frac{15}{16}(1-t^2)^2$
Epanechnikov	$\frac{3}{4}(1-t^2)$
Gaussian	$\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right)$
Triangle	$1- t $
Triweight	$\frac{35}{32}(1-t^2)^3$

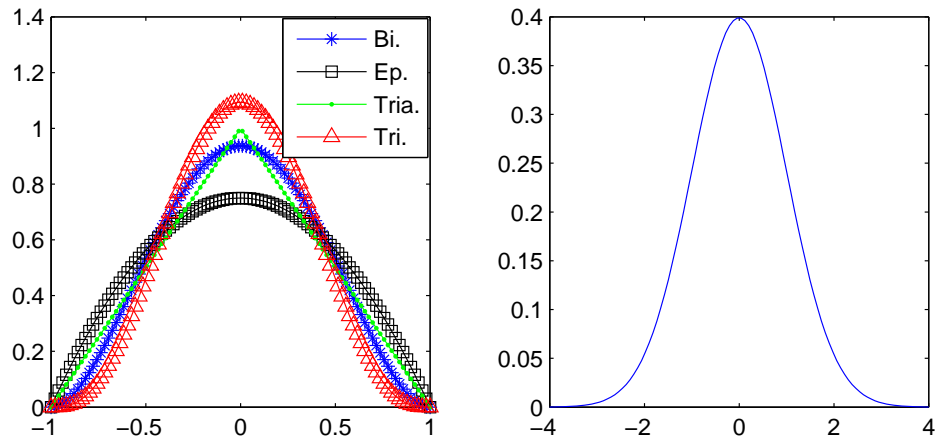


Figure 6.3: Kernel Functions from Table 6.1.

kernel, described by Silverman (1986) and Scott (1992):

$$g(x_i) = \frac{1}{n \prod_{j=1}^p h_j} \sum_{l=1}^n \left[ \prod_{j=1}^p K\left(\frac{x_{lj} - x_{ij}}{h_j}\right) \right], \quad (6.2)$$

where  $x_{lj}$  is the  $j^{\text{th}}$  measurement on the  $l^{\text{th}}$  observation. Clearly, (6.2) is nothing more than the product of  $p$  univariate kernels, where each dimension has its own, possibly different, window width (hence the name). Note that the product kernel assumes all dimensions are independent; this major simplification can often be overly restrictive (more later). Thus, our results are based on a multivariate kernel density estimator using the Gaussian kernel function, shown in (6.3).

$$g(x_i) = \frac{1}{n (2\pi)^{\frac{p}{2}}} \sum_{l=1}^n |H|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (x_l - x_i)' H^{-1} (x_l - x_i)\right) \quad (6.3)$$

The main requirement here is that the bandwidth matrix  $H$  be symmetric  $p \times p$ , positive definite, and non-singular. If  $H$  is diagonal, (6.2) and (6.3) should return identical density estimates.

To demonstrate the flexibility of multivariate kernel density estimation, we generated  $n = 100$  random samples from two skewed univariate PE distributions. If  $g(x)$  is a univariate symmetric distribution centered on  $\mu$ , Fernandez and Steel (1998) suggested the distribution could be skewed by creating a new distribution

$$f(x) = \frac{2}{\kappa + \frac{1}{\kappa}} \left[ g\left(\frac{x - \mu}{\kappa}\right) I(x \geq \mu) + g((x - \mu)\kappa) I(x < \mu) \right], \quad (6.4)$$

where  $\kappa$  controls the amount of skewness. For  $\kappa = 1$ ,  $g(x) = f(x)$  in (6.4). Figure 6.4 demonstrates the skewed PE distribution with  $\beta = 0.5$  using  $\kappa = [-2, 1, 2]$ . For our simulation, both variables were skewed in a different direction and at a different magnitude ( $\kappa = -2, 4$ ), then we used a *copula* with covariance of  $\sigma_{12} = 0.5$  to induce some dependence. In Figure 6.5, we see that the kernel density estimation adapted to the shape of the data quite nicely.

## 6.2 Bandwidth Estimation

Although using kernels frees us from enforcing a functional form, there is no free lunch. While optimizing the kernel function may not provide much value, selecting the right bandwidth matrix does. This can best be demonstrated by considering a simple histogram. In creating a histogram

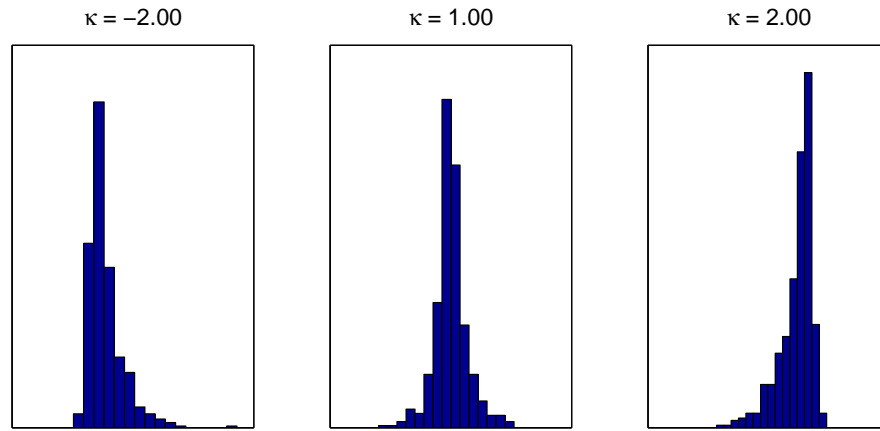


Figure 6.4: Histograms of 3 Examples of the Skewed PE Distribution.

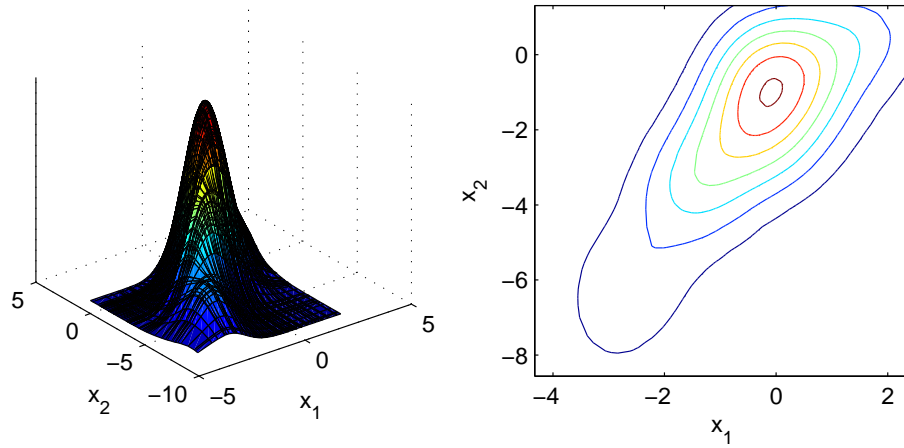


Figure 6.5: Bivariate KDE Surface and Contours of Skewed Data.

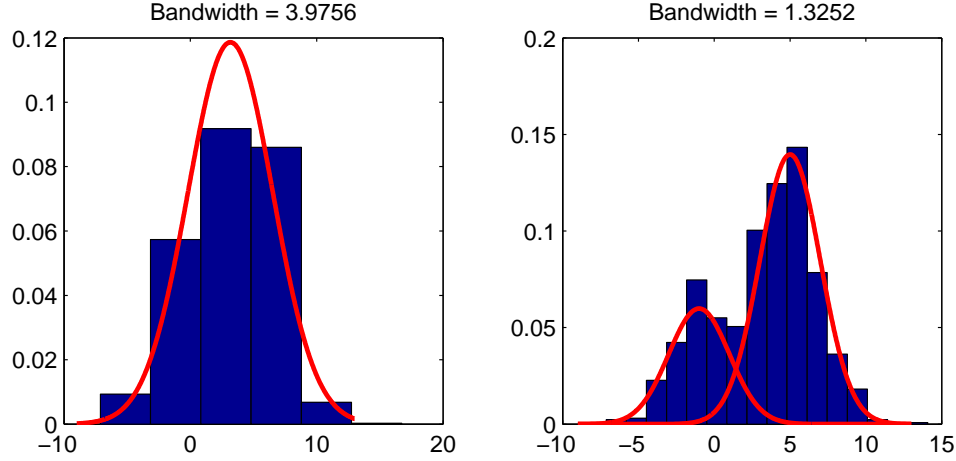


Figure 6.6: Demonstrating the Importance of Appropriate Bandwidth Selection.

of some data, we usually select the number of bands,  $B$ , into which the data will be placed (or we let the computer do so). What’s really occurring, however, is that a bandwidth is being computed as

$$h = \frac{\max(X) - \min(X)}{B}. \quad (6.5)$$

The choice of  $h$  can have a huge impact upon how we view the data analyzed. Figure 6.6 is an excellent demonstration of this fact. We generated a total of  $n = 1000$  random samples from a mixture of Gaussians:

$$X \sim 0.3N(-1, \sigma^2 = 2) + 0.7N(5, \sigma^2 = 2).$$

The two separate densities can be seen superimposed on the histogram in the right panel. In the left panel, the density histogram is computed using  $B = 5$  bins, while  $B = 15$  bins are used in the second. Of course, using  $B = 5$  bins is merely a pedagogical extreme. It is clear that using  $H = 3.98$  “smooths out” the bimodality of the data, while using  $H = 1.33$  helps the researcher identify this characterization. The situation translates directly from histogram density estimation to kernel density estimation - using an inappropriate bandwidth matrix can either conceal too many features in the data, or be overly influenced by them. The obvious solution is to use a data-adaptive computation for  $H$ .

One approach for choosing the value for a univariate  $h$  is to minimize the *asymptotic mean*

integrated squared error (AMISE); for a nonnegative univariate kernel, we have

$$AMISE = \frac{R(K)}{nh} + \frac{1}{4}\sigma_K^4 h^4 R(f''), \quad (6.6)$$

where  $R(\cdot)$  is a function that measures “roughness”, and  $K$  is continuous, centered on 0 with positive variance  $\sigma_K^2$ . Thus, for the Gaussian kernel, we get the *Normal Reference Rule*:

$$h = \left( \frac{R(K)}{n\sigma_K^4 R(f'')} \right)^{\frac{1}{5}} = \left( \frac{4}{3n} \right)^{\frac{1}{5}} \sigma. \quad (6.7)$$

If it provides a smaller value for  $h$ , Silverman (1986) recommends replacing  $\sigma$  with  $\frac{IQR}{1.348}$ , as a robust estimator of the standard deviation. If one wishes to use the Gaussian product kernel for density estimation, the Normal Reference Rule (Scott, 1992) is shown in (6.8).

$$h_j = \left( \frac{4}{n(p+2)} \right)^{\frac{1}{p+4}} \sigma_j, j = 1, \dots, p \quad (6.8)$$

Of course, things get a lot more complicated when we consider multivariate KDE. If we denote  $\mu_2^2$  to be the second moment of the kernel function,  $H_K$  to be its Hessian matrix, and let  $\|K\|_2^2$  be the squared  $L_2$  norm of  $K$ , the AMISE is

$$AMISE = \frac{\mu_2^2}{4} \int [tr(H' H_g H)]^2 dx + \frac{\|K\|_2^2}{n|H|}, \quad (6.9)$$

see Wand and Jones (1995). For bivariate data with the Gaussian kernel, Wand and Jones (1993) derive

$$H = \frac{3}{8\pi} |\Sigma|^{-\frac{1}{2}} n^{-\frac{2}{3}}, \quad (6.10)$$

which they find appealing, since

*“It simply says, that to optimally estimate a bivariate normal density, one should have kernel mass with the same covariance structure as the density itself.”*

This is a logical basis for multivariate extensions of the Normal Reference Rule, in which  $H$  is computed based on the estimated sums-of-squares or covariance matrix.

$$W = \sum_{i=1}^n (x_i - \hat{\mu})' (x_i - \hat{\mu}), \quad \hat{\Sigma} = \frac{1}{n-1} W \quad (6.11)$$

Table 6.2: Bandwidth Matrix Estimation Methods.

Method	$H$ Formula	Description	Parameters Estimated
1	$\frac{1}{np} \text{tr}(W) I_p$	spherical diagonal	1
2	$\frac{1}{n} \text{diag}(W)$	ellipsoidal diagonal	$p$
3	$\frac{1}{n} W$	general	$\frac{p(p+1)}{2}$
4	$\left(\frac{4}{n(p+2)}\right)^{\frac{2}{p+4}} \hat{\Sigma}$	general	$\frac{p(p+1)}{2}$
5	$\left(\frac{4}{n(p+2)}\right)^{\frac{1}{p+4}} \text{diag}(\hat{\Sigma})$	ellipsoidal diagonal	$p$
6	$\left(\frac{4}{n(p+2)}\right)^{\frac{1}{p+4}} C_1(\hat{\Sigma})$	spherical diagonal	1

General shapes for  $H$  include spherical diagonal (identical bandwidth for each dimension), ellipsoidal diagonal (different bandwidth for each dimension), or completely general (different bandwidths with interaction). Six common methods for computing  $H$  are shown in Table 6.2, partially adopted from Bensmail and Bozdogan (2002). Method five reduces to the normal reference rule in (6.7) and (6.8). As the normal reference rule does not consider the correlational structure of the data, method 6 is more general, utilizing the maximal entropic complexity of the entire covariance matrix. In cases of an ill-conditioned or even singular  $W$ , the bandwidth matrix may not be invertible, depending upon the computation method used. When this occurs, we use one of the robust covariance estimators from Chapter 3. However, on occasion,  $H$  could be beyond repair. If the bandwidth matrix for the  $k^{\text{th}}$  becomes uncomputable, the posterior probability of group membership for all datapoints in that group are set to zero, so the datapoints migrate into other populations. With so many ways to estimate the bandwidth matrices  $H_k$ , we need a way to choose which method to use. As with the robust covariance estimators, we suggest to use information criteria for this. We begin by fitting a mixture model of exactly  $K_{\max}$  groups to a given dataset - once for each estimation method in consideration. Whichever method produces the minimum score is selected for use with that dataset in the subsequent complete analysis.

A completely different approach for estimating  $H$  empirically (not for mixture distributions) has been suggested by Zhang et al. (2004). In this research, they suggest an MCMC approach to maximize the cross-validation likelihood criteria; it is claimed to be the "...first practical method for estimating the optimal bandwidth matrix." Though the Markov Chain apparently scales with  $p$  very well, nothing concrete was stated regarding timing. KDE methods are already computationally

intensive; computing kernel density estimates thousands of times (they used a burn-in period of 5000 iterations) and using cross-validation seems mind-bogglingly so. Now consider doing this when trying to fit a mixture model with  $\hat{K} = 1 \dots 6!$  Our  $H_k$  selection approach is intuitive, reasonable, and practical.

### 6.3 Hybrid EM Algorithm for the KMM

When fitting a KDE mixture model, the researcher can choose between estimating a common bandwidth matrix (same  $H$  across all groups), or estimating group-specific bandwidth matrices. In general, we feel that it is best to make as few restrictive decisions in the modeling process as possible - the data should guide us. Therefore, we prefer to employ mixture-specific bandwidth matrices. Thus, for the  $k^{\text{th}}$  group, the density function is

$$g_k(x_i | H_k) = \frac{1}{n (2\pi)^{\frac{p}{2}}} \sum_{l=1}^n |H_k|^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (x_l - x_i) H_k^{-1} (x_l - x_i)' \right), \quad (6.12)$$

and the log-likelihood for the KMM is shown in (6.13).

$$\log L(\hat{\theta} | X) = \sum_{i=1}^n \log \left[ \sum_{k=1}^{\hat{K}} \hat{\pi}_k g_k(x_i | H_k) \right] \quad (6.13)$$

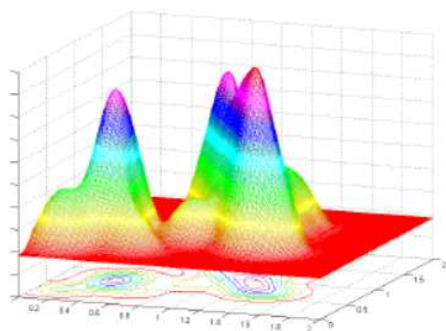
Fitting a mixture of kernel density estimators to data follows the same general process as the mixture of Gaussians. The  $t^{\text{th}}$  iteration of the EM algorithm computes the posterior probability of group membership given by

$$\hat{p}_i(k) = \frac{\hat{\pi}_k^{(t-1)} g_k(x_i | H_k^{(t-1)})}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k^{(t-1)} g_k(x_i | H_k^{(t-1)})}. \quad (6.14)$$

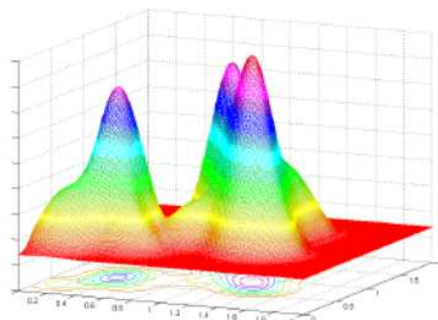
The primary difference here is that the MAP rule must be used for computing the density  $g_k(x_i | H_k^{(t-1)})$ . The datapoints having the highest probability in group  $k$  are used for estimating  $H_k^{(t-1)}$ . Hence, this is another hybrid EM algorithm. Following these computations in the **E**-step, the **M**-step entails re-estimating the mixing proportion (2.10) and covariance matrix (2.12) for each group; the latter is then used to re-estimate the group-specific bandwidth matrices.

As already mentioned, KDE allows us to compute density estimates for some data empirically,

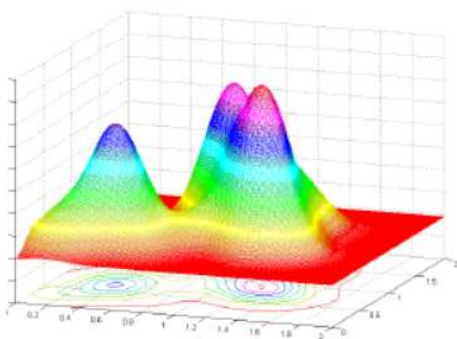
without imposing any form. Thus, we can model characteristics that don't fit any distributions. Another benefit is the smoothing effect of the kernels; the averaging that occurs helps smooth the likelihood surface in fewer iterations than the other mixture models. This has the effect of reducing the overall gradient of the surface and the number of local maxima. Figures 6.7a through 6.7d demonstrate the value of this smoothing very well (Reddy and Chiang, 2007).



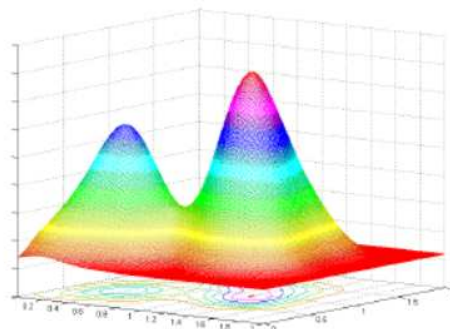
(a) The Original Log-likelihood Surface Which is Very Rugged.



(b) The Intermediate Smoothed Surface.



(c) The Intermediate Smoothed Surface.



(d) Final Smoothed Surface with Only Two Local Maxima.

Figure 6.7: Evolution of Likelihood Surface Using KDE Smoothing.



## 6.4 Information Criteria for the KMM

For the KMM, the number of parameters  $m$  is a function of how  $H_k$  is estimated. For example, using method 5,  $m = \hat{K}p + (\hat{K} - 1)$ ;  $\hat{K} - 1$  mixing proportions, and the diagonal elements of each mixture-specific bandwidth matrix. There are no changes to the forms of either *AIC* or *SBC*. When employing a mixture of kernels model, the IFIM in (3.31) remains virtually unchanged, with the exception that  $\hat{\Sigma}_k$  is replaced with  $H_k$ :

$$\hat{\mathcal{F}}_k^{-1} = \begin{bmatrix} H_k & \mathbf{0} \\ \mathbf{0} & \left(\frac{2}{n_k}\right) D_p^+ (H_k \otimes H_k) D_p^{+'} \end{bmatrix}. \quad (6.15)$$

The situation is identical for *ICOMP*; in (6.16),  $h_{kjj}$  indicates the bandwidth for dimension  $j$  in the  $k^{\text{th}}$  cluster.

$$\begin{aligned} \text{ICOMP}(\hat{\mathcal{F}}^{-1}) &= -2 \log L(\hat{\theta} \mid X) \\ &+ m \left( \log \left[ \sum_{k=1}^{\hat{K}} \left\{ \frac{\text{tr}(H_k)}{\hat{\pi}_k} + \frac{1}{2} \left( \text{tr}(H_k^2) + \text{tr}(H_k)^2 + 2 \sum_{j=1}^p h_{kjj}^2 \right) \right\} \right] \right. \\ &\left. - \log m \right) - \left\{ (p+2) \sum_{k=1}^{\hat{K}} \log |H_k| - p \sum_{k=1}^{\hat{K}} \log(\hat{\pi}_k n) \right\} - \hat{K} p \log(2n) \end{aligned} \quad (6.16)$$

As before, the modification required to compute  $\text{ICOMP}_{PEU}$  instead is straightforward.

# The Power Exponential Kernel Mixture Model (PEKMM)

*“Welcome everyone, I’m your dam tour guide Arnie. I’m about to take you through a fully functional **power** plant. So please no one wander off the dam tour, and feel free to take all the dam pictures you want. Now, are there any dam questions?”- Arnie, National Lampoon’s Vegas Vacation*

## 7.1 Power Exponential Product Kernel

As we saw in Chapter 5, the multivariate power exponential distribution is a special case of the Kotz Type elliptically-contoured class of distributions. In the PE density function, shown in (7.1)

$$f(x_i | \mu, \Sigma, \beta) = \frac{p \Gamma\left(\frac{p}{2}\right) |\Sigma|^{-\frac{1}{2}}}{2\pi^{\frac{p}{2}} \Gamma\left(1 + \frac{p}{2\beta}\right) 2^{1+\frac{p}{2\beta}}} \exp\left(-\frac{1}{2} [(x_i - \mu) \Sigma^{-1} (x_i - \mu)]^\beta\right), \quad (7.1)$$

note that the shape parameter  $\beta$  is used for the entire dataset - all dimensions are forced to conform to the same tail behavior. In real datasets, however, it can be valuable to allow for different tail/peak behavior by variable. For example, we simulated a dataset with  $n = 500$  observations of  $p = 3$  variables independently generated from the univariate PE, with  $\beta_1 = 0.5$ ,  $\beta_2 = 1.0$ , and  $\beta_3 = 10.0$ . Using the estimation protocol given in Chapter 5 for the Kotz subclass, we compute  $\hat{\beta} = 0.280$ ; interestingly, this approach was unable to even provide a result that was near the average of the true values. When computed the MoM estimator using (5.14) on a dimension-by-dimension

Table 7.1: Maximized Log-likelihoods.

PE Distribution	-175265.77
Multivariate Gaussian Kernel	-1538.30
<b>PE Product Kernel</b>	<b>-1444.75</b>

basis, however, we get  $\hat{\beta}_1 = 0.471$ ,  $\hat{\beta}_2 = 0.928$ , and  $\hat{\beta}_3 = 10.0$  - all very accurate. By estimating a shape parameter for each variable, we may get a much better fit to the data. For this simulated dataset, we fit three models and computed the maximized likelihoods, with the results shown here in Table 7.1. We see that the PE distribution, assuming all dimensions had the same shape, did not provide the best fit, according to the likelihood. The multivariate Gaussian kernel (using the normal reference rule for  $H$ ) fit the data dramatically better, but the power exponential product kernel (PEPK) fit the best.

Thus, we introduce the PE product kernel, as it allows us to model the tail/peak behavior in each dimension *individually* and *independently*. For the  $i^{\text{th}}$  observation, the estimated density is computed using (6.2), repeated here.

$$g(x_i) = \frac{1}{n \prod_{j=1}^p h_j} \sum_{l=1}^n \left[ \prod_{j=1}^p K\left(\frac{x_{lj} - x_{ij}}{h_j}\right) \right] \quad (7.2)$$

Combining (7.1) with (7.2), we define the power exponential kernel for the  $j^{\text{th}}$  variable as

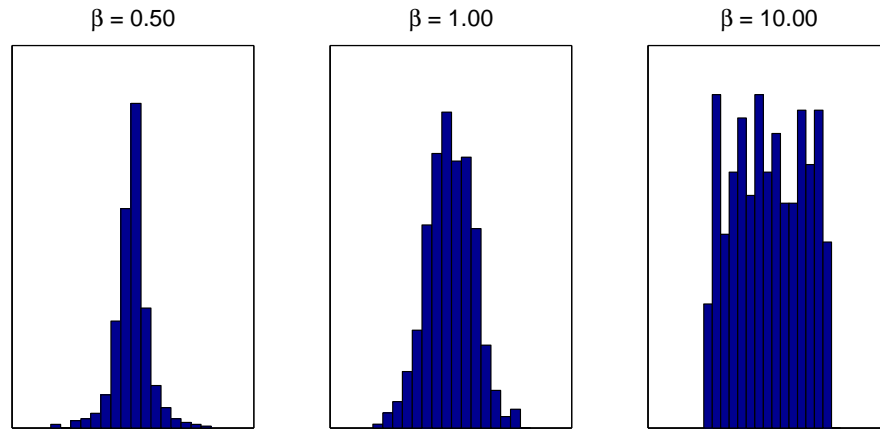
$$K(t_i) = \left[ \Gamma\left(1 + \frac{1}{2\beta_l}\right) 2^{1+\frac{1}{2\beta_l}} \right]^{-1} \exp\left(-\frac{1}{2} t_i^{2\beta_l}\right), \quad (7.3)$$

where  $t = (x_{lj} - x_{ij}) / h_j$ . The bandwidth values are computed using method 5 in Table 6.2:

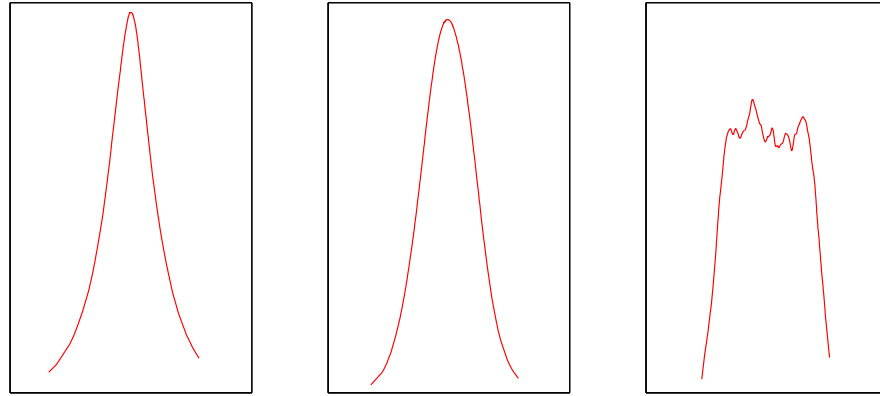
$$h_j = \left( \frac{4}{n(p+2)} \right)^{\frac{1}{p+4}} \hat{\sigma}_j^2. \quad (7.4)$$

From the already mentioned simulation, we have the histograms and density estimates in Figure 7.1. This allows us to visualize why the PEPK model fit the data so much better than the others - the shape of each dimension is fit individually.

Of course, there are always trade-offs to make in any modeling procedure. In fitting the PE



(a) Histograms of 3 Examples of Univariate PE Data.



(b) Estimated Densities of 3 Examples of Univariate PE Data.

Figure 7.1: Fitting PE Univariate Kernel to Each Dimension of Simulation.

product kernel to a dataset, we lose the ability to correctly model the information in the dependency between the variables. To demonstrate this, we went back to our simulation protocol and induced moderate correlation using a copula with the covariance matrix

$$\Sigma = \begin{bmatrix} 1.0 & 0.5 & -0.5 \\ 0.5 & 0.75 & 0.0 \\ -0.5 & 0.0 & 2.0 \end{bmatrix}.$$

When the PE product kernel was used on this correlated dataset, it performed very poorly at recovering the true  $\beta$  parameters - it got  $[0.471, 0.555, 0.276]$ . Additionally, the log-likelihood scores indicated that the multivariate Gaussian kernel fit the data better ( $-2127.43$  vs.  $-3257.78$ ). One solution would be to replace  $\hat{\sigma}_j$  in the computation for  $h_j$  with some scalar measure which makes use of the entire covariance matrix -  $c_1(\hat{\Sigma})$ , for example. Of course, this would then enforce a common bandwidth for all dimensions, which may not be desirable. In summary, it seems important to decide whether we prefer to better fit the distributional shape or dependency structure in a given dataset.

## 7.2 Hybrid EM Algorithm for the PEKMM

For the PEPK mixture model, the **E**-step of the EM algorithm computes the posterior probability of group membership shown in (7.5)

$$\hat{p}_i(k) = \frac{\hat{\pi}_k^{(t-1)} g_k(x_i | h_k^{(t-1)}, \hat{\beta}_k^{(t-1)})}{\sum_{k=1}^{\hat{K}} \hat{\pi}_k^{(t-1)} g_k(x_i | h_k^{(t-1)}, \hat{\beta}_k^{(t-1)})}, \quad (7.5)$$

where  $h_k = [h_1, \dots, h_p]$  and  $\hat{\beta}_k = [\hat{\beta}_1, \dots, \hat{\beta}_p]$ . The probability density  $g(x)_k$  for the  $i^{\text{th}}$  observation in the  $k^{\text{th}}$  group is shown in (7.6).

$$g_k(x_i) = \frac{1}{n \prod_{j=1}^p h_{jk}} \sum_{i=1}^n \left[ \prod_{j=1}^p \left\{ \Gamma \left( 1 + \frac{1}{2\hat{\beta}_{jk}} \right) 2^{1+\frac{1}{2\hat{\beta}_{jk}}} \right\}^{-1} \exp \left( -\frac{1}{2} \left( \frac{X_j - x_{ij}}{h_{jk}} \right)^{2\hat{\beta}_{jk}} \right) \right] \quad (7.6)$$

At the  $t^{\text{th}}$  iteration, the **M**-step entails computing the following quantities:

$$\hat{\pi}_k^{(t)} = \frac{1}{n} \sum_{i=1}^n \hat{p}_i(k), \quad (7.7)$$

$$\hat{\mu}_k^{(t)} = \frac{1}{n\hat{\pi}_k^{(t)}} \sum_{i=1}^n x_i \hat{p}_i(k), \quad (7.8)$$

$$\hat{\Sigma}_k^{(t)} = \frac{1}{n\hat{\pi}_k^{(t)}} \sum_{i=1}^n \hat{p}_i(k) (x_i - \hat{\mu}_k^{(t)})' (x_i - \hat{\mu}_k^{(t)}), \quad (7.9)$$

$$h_k^{(t)} = \left( \frac{4}{n\hat{\pi}_k^{(t)}(p+2)} \right)^{\frac{1}{p+4}} \text{diag}(\hat{\Sigma}_k^{(t)}). \quad (7.10)$$

For estimating the kurtosis parameter of the  $j^{\text{th}}$  variable in the  $k^{\text{th}}$  mixture, the average squared Mahalanobis distance is

$$\overline{d^2}_{jk} = \frac{1}{n\hat{\pi}_k^{(t)}\hat{\sigma}_{jk}^{4(t)}} \sum_{i=1}^n \hat{p}_i(k) (x_{ij} - \hat{\mu}_{jk}^{(t)})^4, \quad (7.11)$$

where  $\hat{\mu}_{jk}^{(t)}$  indicates the mean of the  $j^{\text{th}}$  variable, and  $\hat{\sigma}_{jk}^{4(t)}$  the square of its variance. Finally, the root (using the implementation method from Chapter 5) of (7.12) is the estimated kurtosis parameter.

$$\frac{\Gamma\left(\frac{1}{2\hat{\beta}_{jk}^{(t)}}\right) \Gamma\left(\frac{5}{2\hat{\beta}_{jk}^{(t)}}\right)}{\Gamma^2\left(\frac{3}{2\hat{\beta}_{jk}^{(t)}}\right)} - \overline{d^2}_{jk} \quad (7.12)$$

### 7.3 Information Criteria for the PEKMM

Recall that the product kernel is a simplistic way to compute a multivariate kernel density estimate by considering each dimension independently. In each dimension, we must estimate  $\mu$ ,  $h$ , and  $\beta$ . Thus, the number of parameters estimated for the PEPK mixture model is

$$m = \hat{K}(3p+1) - 1. \quad (7.13)$$

*AIC* and *SBC* are computed as shown in (7.14) and (7.15).

$$AIC = -2 \sum_{i=1}^n \log g(x_i) + 3m \quad (7.14)$$

$$SBC = -2 \sum_{i=1}^n \log g(x_i) + \log(n) m \quad (7.15)$$

For this mixture model, *ICOMP* is even easier to compute than for the others. In Chapter 6, (6.16) showed how to compute *ICOMP* for the KMM, requiring only traces and determinants. The trace and determinant of a diagonal matrix

$$D = \begin{bmatrix} d_{11} & & & \mathbf{0} \\ & d_{22} & & \\ & & \ddots & \\ \mathbf{0} & & & d_{pp} \end{bmatrix},$$

is simply the sum and product, respectively, of the diagonal elements. Thus, *ICOMP* and *ICOMP*<sub>PEU</sub> for this mixture model are shown in (7.16) and (7.17).

$$\begin{aligned} ICOMP(\hat{\mathcal{F}}^{-1}) = & -2 \sum_{i=1}^n \log g(x_i) + m \log \left[ \frac{1}{m} \sum_{k=1}^{\hat{K}} \left\{ \frac{tr(h_k)}{\hat{\pi}_k} + \frac{1}{2} (3tr(h_k^2) + tr(h_k)^2) \right\} \right] \\ & - \left\{ (p+2) \sum_{k=1}^{\hat{K}} \log |h_k| - p \sum_{k=1}^{\hat{K}} \log(\hat{\pi}_k n) \right\} - \hat{K} p \log(2n) \end{aligned} \quad (7.16)$$

$$\begin{aligned} ICOMP(\hat{\mathcal{F}}^{-1})_{PEU} = & -2 \sum_{i=1}^n \log g(x_i) + \left\{ m \log \left[ \frac{1}{m} \sum_{k=1}^{\hat{K}} \left\{ \frac{tr(h_k)}{\hat{\pi}_k} + \frac{1}{2} (3tr(h_k^2) + tr(h_k)^2) \right\} \right] \right. \\ & \left. - \left\{ (p+2) \sum_{k=1}^{\hat{K}} \log |h_k| - p \sum_{k=1}^{\hat{K}} \log(\hat{\pi}_k n) \right\} - \hat{K} p \log(2n) \right\} \frac{\log n}{2} + m \end{aligned} \quad (7.17)$$

## Numerical Results

*“IT... COULD... **WORK!**”* - Dr. Frankenstein, Young Frankenstein

All results in this chapter were obtained using our  $M^3$  toolbox, described in the appendix. In the following studies, we used the GA parameters shown below, fitting  $\hat{K} = 1 \dots 6$  for all datasets. Recall from Chapter 4 that heuristic guidelines for GA parameters settings are few and far between. Thus, the selection of specific values must include a level of arbitrariness. Most of the settings chosen here were selected by observing convergence behavior of the GA with a variety of datasets, and in a variety of settings. We opted not to use Elitism, because of the way it increases the time required for each generation. Finally, recall the population size heuristic for variable subsetting -  $P > p$ . Logical generalization of this to the mixture problem would suggest that population size should scale with  $n$ . A single generation could take a very long time for a large dataset, and risk losing important results, in the event of a software / hardware crash in the middle of a generation. Rather than set the population size very high, we prefer to use a moderate value, and possibly perform several replications of the GA. This should allow us to still search the parameter space well (possibly even better) without risking too much computation time on a single replication.

- Number Generations - 60
- Premature Termination Threshold - 40
- Population Size - 30
- Generation Seeding - Roulette
- Crossover Probability - 0.75
- Mutation Probability - 0.10
- Elitism - Off



Where the EM or K-Means algorithms were used, the convergence criteria was set to  $C = 10^{-6}$ , and a maximum of 1000 iterations were allowed. All datasets are described in more detail in the appendix. Finally, in tables recording model selection frequencies, we indicate the true structure with **bold** typeface, and the models selected by each criteria are indicated by an \* next to the score.

As we present and discuss model selection results, we judge models based on the known true class labels. For example, we may have a dataset from a medical study in which patients are classified as either *sick* or *healthy*. A criteria that selects a model with four populations is said to overfit; a model with a single population is said to be underfit. Any observations that are placed in an incorrect or spurious population are counted as misclassified. While this is consistent with the concept of classification, it can be considered somewhat unnatural. Underlying this way of measuring a model is the assumption that the class labels accurately reflect all relevant information. Using our medical example, it could be that a more accurate grouping structure would be *sick male*, *sick female*, *healthy male*, and *healthy female*. If a procedure selected the  $\hat{K} = 4$  model correctly identifying this structure, we would say it overfit and that half (assuming equally-sized groups) of the observations were misclassified. We could, in fact, justify the use of mixture modeling to characterize inhomogeneities within known grouping structures. However, we are limited by the fact that we would have no **easily-interpretable**, **justifiable**, and **objective** measure of model performance. In fact, we would have no certain measure of model performance, since we could argue any given model was identifying some ethereal characteristic the others weren't (and that we ourselves couldn't). While we briefly make observations along these lines for a few upcoming datasets, we rely upon correct classification as an objective measure of model performance.

## 8.1 Traditional GMM

We begin by showing some numerical results without (mostly) using any of the newer techniques, for the purpose of comparison. These same datasets will be reanalyzed in Sections 8.2 through 8.5 with much better results than what is shown here.

### 8.1.1 Simulation S1 - Mixed Overlapping

Our first simulation study is using the dataset previously shown in Figure 2.1. This data was generated using simulation protocol S1, shown in the appendix, with  $n = 300$  observations - 100

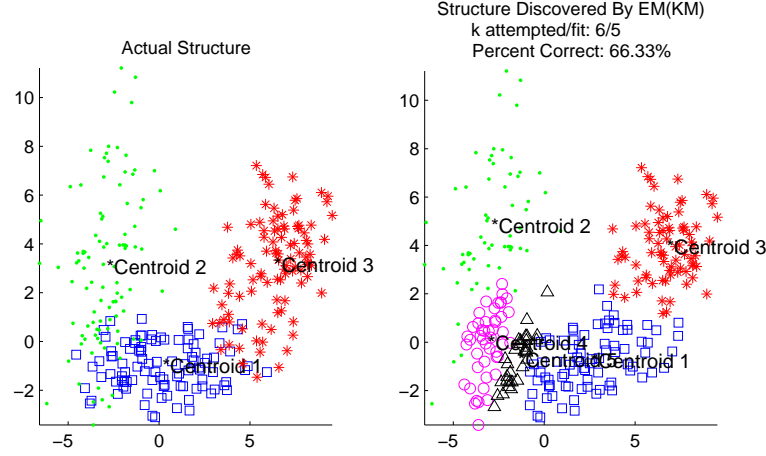


Figure 8.1: Simulation S1 - Results from Best Model as Selected by Log-likelihood.

from each of  $K = 3$  groups. For this moderately overlapped dataset, these methods were unable to produce consistent results. The EM algorithm was able to converge for  $\hat{K} = 2$  in 34 of the  $M = 100$  simulations; the number was even worse when trying to fit a model with  $\hat{K} = 3$  mixtures - 23. In fact the EM(KM) produced results for  $\hat{K} = 1 \dots 6$  in only 1 simulation! The modeling results from this single simulation are presented in Table 8.1. Throughout this section, we use \* in the model selection frequency tables to indicate the model structure at which criteria are minimized. Note the fifth row - when the EM algorithm attempted to fit  $\hat{K} = 5$  mixtures, one of them dropped out, and it replicated the results when attempting to fit  $\hat{K} = 4$  Gaussians to the data. We see that the likelihood was maximized for  $\hat{K} = 5$  groups, even though it is on a boundary. Perhaps the best model would have more groups, but given the convergence problems with this highly overlapped dataset, the chances of getting an answer for higher  $K_{\max}$  seem slim. In the right pane of Figure 8.1, we see how the  $K = 3$  actual groups were split into five.

Table 8.1: Simulation S1 - Results from Sole Completed Simulation.

$\hat{K}$ attempted,fit	$\log L(\hat{\theta}   X)$	Correct Classification Rate
1, 1	-1602.87	33.33
2, 2	-1496.06	66.67
<b>3, 3</b>	<b>-1455.73</b>	<b>88.33</b>
4, 4	-1448.35	69.00
5, 4	-1448.35	69.00
6, 5*	-1439.32	66.33

### 8.1.2 Simulation S3 - Spherical Overlapping

Next we have a simulation protocol with  $n = 150$  observations created from  $K = 3$  evenly-sized spherical mixtures in which the third population is almost entirely contained in the second. Visual inspection of the left panel in Figure 8.2 (without different colors/markers per group) shows that  $\hat{K} = 2$  is a likely candidate for the number of populations. Only the higher density in part of the upper cluster suggests there are actually three. As humans, we can identify this discrepancy - the question is “*Can the computer do so?*”. We performed  $M = 100$  simulations from this protocol, attempting to fit the six possible mixture models to this data with EM(KM). In only 17 simulations did the EM algorithm converge to a solution for all six attempted models. For all of these, the single-population model was chosen as the best. In Table 8.2, we show the results from the simulation with the best overall results. For the model with the correct structure, we see that only 6 observations were misclassified - not too shabby.

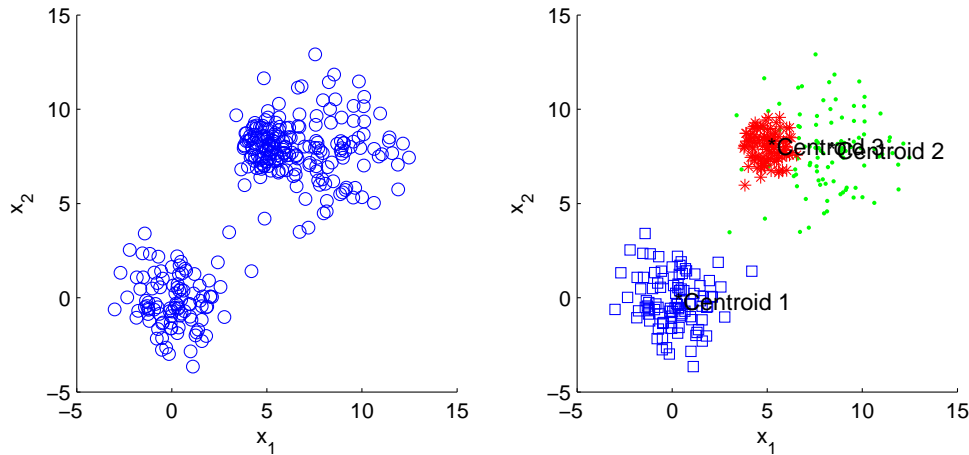


Figure 8.2: Simulation S3 - Sample Scatter Plot of  $X_1$  Against  $X_2$ .

Table 8.2: Simulation S3 - Model Selection Results from Best Simulation With Complete Convergence.

$\hat{K}$	$\log L(\hat{\theta}   X)$	Correct Classification Rate
1*	763.71	33.33
2	681.24	66.67
<b>3</b>	<b>635.32</b>	<b>96.00</b>
4	630.03	90.00
5	621.55	90.00
6	615.92	79.33

Table 8.3: Iris data - Gaussian Mixture Model Selection Results.

$\hat{K}$	$\log L(\hat{\theta}   X)$	Correct Classification Rate
1*	379.91	33.33
2	214.35	66.67
<b>3</b>	<b>180.35</b>	<b>96.67</b>
4	—	—
5	—	—
6	—	—

### 8.1.3 Real data - Iris

The next dataset is Fisher’s iris data. This dataset consists of  $p = 4$  flower characteristics: *petal length*, *petal width*, *sepal length*, and *sepal width*. There are  $K = 3$  groups: 50 observations each from the varieties *Iris Setosa*, *Iris Versicolor*, and *Iris Virginica*. For  $\hat{K} = 1 \dots 6$ , we executed the EM algorithm, initialized by K-Means. Model likelihood scores and classification rates are shown in Table 8.3. The EM algorithm was only able to converge to a solution when fitting a bi-group or tri-group structure. Considering just the models for which results were obtained, we see that maximizing the likelihood would lead to the conclusion that the data represent a single population. That said, when fitting the  $\hat{K} = 3$  model, only 5 observations were misclassified.

### 8.1.4 Real data - Aorta

The second real dataset used in this section is medical imaging data from a study of heart tissue. Hardening of the arteries is the leading cause of death and debility in the industrial world. In the U.S. alone 13 million Americans suffer from heart attacks, and 90,000 people die from heart disease annually. Nuclear magnetic resonance (NMR) imaging has been used to identify fatty tissues in the arteries to aid in early detection of heart attacks. We have data on  $n = 418$  patients, and  $p = 20$  image variables. There are two classes - patients exhibiting *early atheroma*, and those who were *healthy*. As can be seen in the appendix, this dataset shows a lot of overlap and has extreme nonnormal characteristics.

Once again, we used the K-Means algorithm to initialize the EM; each replication required just 4 seconds to fit all  $\hat{K} = 1 \dots 6$  models. Unlike the simulated example just shown, the EM algorithm consistently converged. In all replications, maximizing the log-likelihood led to putting all datapoints in four groups - a correct classification rate of 77.03%. Table 8.4 shows the output

Table 8.4: Aorta data - Results from a Typical Replication.

$\hat{K}$ attempted,fit	$\log L(\hat{\theta}   X)$	Correct Classification Rate
1, 1	-25994.83	53.59
<b>2, 2</b>	-22786.47	54.07
3, 3	-19951.93	65.55
4, 3	-19551.93	65.55
5, 4*	-16551.66	77.03
6, 3	-19951.93	65.55

from a typical replication. Note that even for the correct structure with two groups, just over half of the datapoints were correctly classified.

### 8.1.5 Real data - Diabetic

For our penultimate example using the traditional methods, we have a dataset that is composed of  $K = 3$  types of patients from a diabetes study. Five medical measurements relating to insulin usage were taken on  $n_1 = 33$  *overt diabetic*  $n_2 = 36$  *chemical diabetic*, and  $n_3 = 76$  *non-diabetic* patients. Due to the clinical similarity between the latter two groups, finding either  $\hat{K} = 2$  or  $\hat{K} = 3$  is acceptable for this dataset. A model with  $\hat{K} = 4$  distinct populations was chosen as best, with a correct classification rate of 77.24%. It is interesting to note that the EM algorithm was consistently unable to converge to a solution for  $\hat{K} = 2$  - one of the two acceptable models. We end with a summary of the results from a single replication in Table 8.5.

Table 8.5: Diabetic data - Results from a Typical Replication.

$\hat{K}$ attempted,fit	$\log L(\hat{\theta}   X)$	Correct Classification Rate
1, 1	-3219.81	52.41
<b>2</b>	unable to converge	
<b>3, 3</b>	<b>-2970.26</b>	<b>66.21</b>
4, 3	-2937.17	86.21
5, 4*	-2906.35	77.24
6, 4	-2909.11	76.55

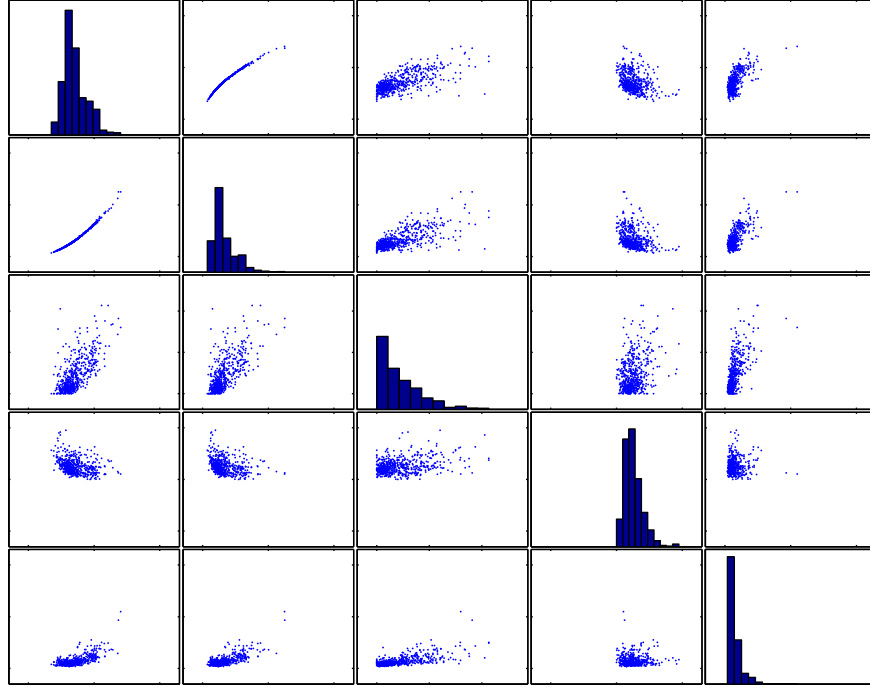


Figure 8.3: Cancer data - Scatter Plot Matrix of Variables a) Through e).

### 8.1.6 Real data - Cancer

Finally, we have a 3<sup>rd</sup> medical dataset first used in Street et al. (1993). Features are computed from a digitized image of a fine needle aspirate of a breast mass from  $n = 569$  patients. They describe characteristics of the cell nuclei present in the image, with  $p = 30$  features. This data is composed of  $K = 2$  groups;  $n_1 = 212$  patients had *malignant* tumors, while the masses of the other  $n_2 = 357$  were *benign* (noncancerous). Figure A.10 from the appendix is repeated in Figure 8.3, without showing the actual groups. These scatter plots show no clear separation between the two populations. We applied the EM algorithm to this dataset, initialized by K-Means. The EM algorithm failed to produce any results, due to covariance singularity, so we allowed the EM to use the convex sum estimator to regularize the estimated covariance matrices when required. This allowed the procedure to consistently converge to a solution for  $\hat{K} = 2$ . For  $\hat{K} = 3 \dots 6$ , however, the algorithm still was unable to converge. For the mixture model with two groups, the algorithm correctly classified 63.1% of the observations, with the resulting confusion matrix shown

Table 8.6: Cancer data - Confusion Matrix from Best Mixture of Normals Model.

		Predicted		
		$k$		Total
Actual	1	2	210	<b>212</b>
	2	0	357	<b>357</b>
	Total	<b>2</b>	<b>567</b>	<b>569</b>

in Table 8.6. Note that these results aren't exactly useful, since we can't identify a best fitting model structure - we used the *a priori* information that the correct structure was  $K = 2$  groups; hence, this result kind of relied upon supervised learning. More important is the fact that this model resulted in a 99% false negative rate; all but two malignant tumors were classified as benign.

## 8.2 Updated GMM

In this section, we apply the updated Gaussian mixture model to various simulated and real-world datasets. We introduce GKM, GARM, GEM, and robust covariance estimation.

### 8.2.1 Simulation S1 - Mixed Overlapping

Our first simulation study is using the dataset shown as a demonstration in Chapter 2. We ran  $M = 50$  simulations with  $n = 300$  using the mixture of Gaussians with GARM to initialize GEM, and using the Empirical Bayes robust covariance estimator. Because of the overlap, there was a lot of variability in the results; Table 8.7 shows the model hit rates. In these results, we make two interesting observations. First of all, note that the consistent criteria were more accurate than the others - they both picked the true structure as the best. However, note that both  $AIC$  and  $ICOMP$  were more precise. They both picked the correct structure with the highest frequency. In Table 8.8 we show the confusion matrix for the best model picked by  $ICOMP_{PEU}$ , which misclassified 28 of the observations. We also show a scatter plot of the true structure (left pane) and estimated structure (right pane) from the best simulation, in Figure 8.4.

Table 8.7: Simulation S1 - Model Selection Frequencies and Results for GEM(GARM).

$\hat{K}$	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
1	0	0	0	0
2	8	32	0	24
<b>3</b>	<b>68</b>	<b>62*</b>	<b>68</b>	<b>52*</b>
4	22*	6	24*	22
5	2	0	6	2
6	0	0	2	0
Correct Classification Rate	85.00%	90.33%	84.33%	90.67%

Table 8.8: Simulation S1 - Confusion Matrix from Best Simulation.

		Predicted				
		$k$	1	2	3	Total
Actual	1	98	1	1		100
	2	17	83	0		100
	3	9	0	91		100
	Total	124	84	92		300



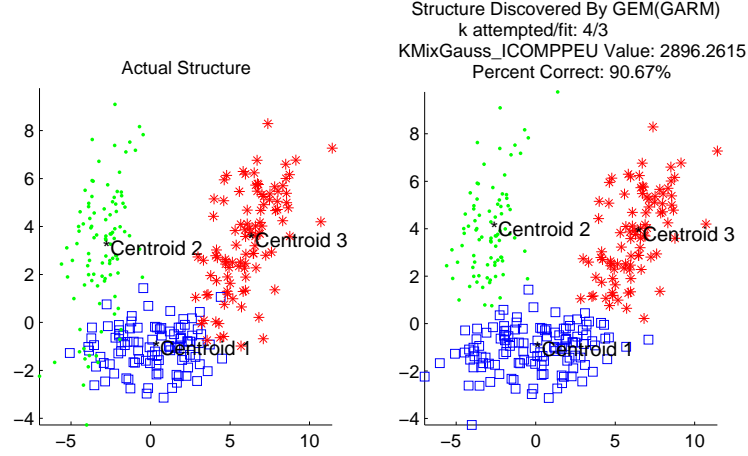


Figure 8.4: Simulation S1 - Scatter Plot of Best Model Structure.

**SUMMARY:** For the GMM using GEM(GARM), we ran 50 simulations using 300 observations.  $ICOMP_{PEU}$  was minimized at a model correctly identifying the true structure of three populations, with a correct classification rate of 90.67%.

### 8.2.2 Simulation S2 - Ellipsoidal Overlapping

Next, we have a dataset created with  $K = 3$  overlapping highly-ellipsoidal bivariate Gaussian mixtures. A small scale Monte-Carlo simulation was performed using this data and  $n = 150$  observations, evaluating the fit of up to  $K_{\max} = 6$  mixtures. The Empirical Bayes covariance smoother (3.40) was utilized for this dataset. No matter which of the three initialization methods in the toolbox were used, the EM algorithm did not consistently converge within 1000 iterations for  $\hat{K} = 1 \dots K_{\max}$ . In fact, it was frequently unable to converge for  $\hat{K} > 3$ . In two of the simulations (initialized by GKM), however, we were able to obtain convergence and estimates for all models up to  $\hat{K} = 4$ , shown in Table 8.9. Note that these results include the turning point of  $ICOMP$ , which justifies their inclusion here. Figure 8.5 demonstrates the dramatic decrease in the  $ICOMP$  scores (and rise in classification rates) from the two-cluster to the three-cluster models. Finally, Figure 8.6 compares the actual and estimated group structures.

Table 8.9: Simulation S2 - Results for Mixture of Gaussians Using EM(GKM).

$\hat{K}$	$ICOMP$	Correct Classification Rate
1	613.87	50.00
2	614.91	69.33
<b>3*</b>	<b>489.43</b>	<b>95.33</b>
4	516.71	83.33

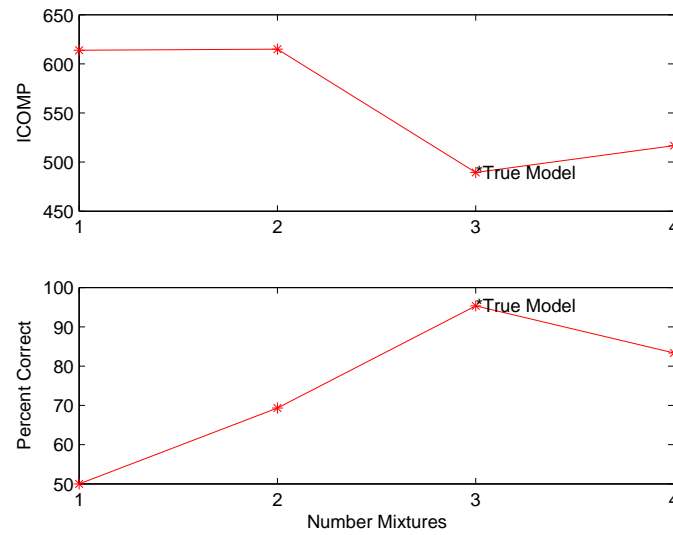


Figure 8.5: Simulation S2 -  $ICOMP$  and Correct Classification Rate Measurements per Model.

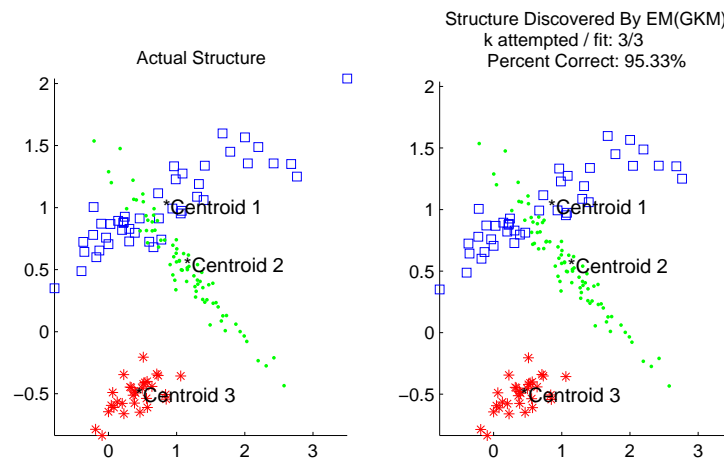


Figure 8.6: Simulation S2 - Scatter Plot of Best Model Structure.

**SUMMARY:** We fit the GMM using EM(GKM) to 150 observations. Most simulations exhibited severe convergence limitations. In one simulation where EM converged for  $\hat{K} = 1 \dots 4$ , *ICOMP* was minimized with a model correctly identifying the true structure of three populations, with a correct classification rate of 95.33%.

### 8.2.3 Simulation S3 - Spherical Overlapping

Here we consider the simulation protocol already analyzed, composed of  $K = 3$  spherical populations. We used this simulation protocol (parameters shown in the appendix) to perform simulation experiments with  $n = 150$  observations. Using GKM to provide initial estimates to the EM algorithm for the mixture of Gaussians (again using  $\hat{\Sigma}^* = \hat{\Sigma}_{MLE/EB}$ ), we executed many Monte-Carlo simulations to obtain one in which convergence was obtained for  $\hat{K} = 1 \dots 6$ . Summary results are shown in Table 8.10. All four information criteria either selected  $\hat{K} = 3$  or  $\hat{K} = 4$ ; none of them were fooled by the “supermixture” even though cluster 3 appears to be part of the second mixture. Note that for this dataset, the heavier penalty enforced by the consistent criteria, *SBC* and *ICOMP*<sub>PEU</sub> was beneficial. That said, no matter which of the two models were selected, the results are phenomenal; the confusion matrix for each model selected is shown in Table 8.11. For the  $\hat{K} = 3$  model, two datapoints were traded between mixtures 2 and 3; considering the level of overlap between them, this 98.67% level of accuracy is remarkable.

Table 8.10: Simulation S3 - Mixtures of Gaussians with EM(GKM) Results.

$\hat{K}$	Correct Classification Rate	<i>AIC</i>	<i>SBC</i>	<i>ICOMP</i>	<i>ICOMP</i> <sub>PEU</sub>
1	33.33	1541.94	1552.00	1535.20	1552.62
2	66.67	1389.95	1412.07	1360.77	1377.53
<b>3</b>	<b>98.67</b>	<b>1335.75</b>	<b>1369.93*</b>	<b>1295.81</b>	<b>1329.47*</b>
4	93.33	1334.28*	1380.52	1283.43*	1333.77
5	82.00	1336.99	1395.30	1284.89	1366.44
6	74.67	1340.41	1410.78	1289.34	1405.53

Table 8.11: Simulation S3 - Confusion Matrices.

$\hat{K} = 3$		Predicted			
Actual	$k$	1	2	3	Total
	1	50	0	0	<b>50</b>
	2	0	49	1	<b>50</b>
	3	0	1	49	<b>50</b>
	Total	<b>50</b>	<b>50</b>	<b>50</b>	<b>150</b>

$\hat{K} = 4$		Predicted				
Actual	$k$	1	2	3	4	Total
	1	50	0	0	0	<b>50</b>
	2	0	41	1	8	<b>50</b>
	3	0	1	49	0	<b>50</b>
	Total	<b>50</b>	<b>42</b>	<b>50</b>	<b>8</b>	<b>150</b>

The model with 4 mixtures had the same trade, and also classified 8 datapoints from mixture 3 into a 4<sup>th</sup> mixture. It is instructive to consider the scatter plot from this model, displayed in Figure 8.7. Note that the eight datapoints in mixture 4 have a substantial degree of separation from the main body of the second mixture, in which they belong. Using the identified mixture model and  $ICOMP_{PEU}$ , we performed influential detection analysis on the data from this simulation. Three observations from group two were flagged as influential, as was one observation from group one. As shown in Figure 8.8, these four observations are clearly separated from the rest. Finally, Table 8.12 has the estimated and actual parameters from this simulation. Results from the “incorrect” model are shown, in order to demonstrate how similar the estimates are, despite the spurious 4<sup>th</sup> group.

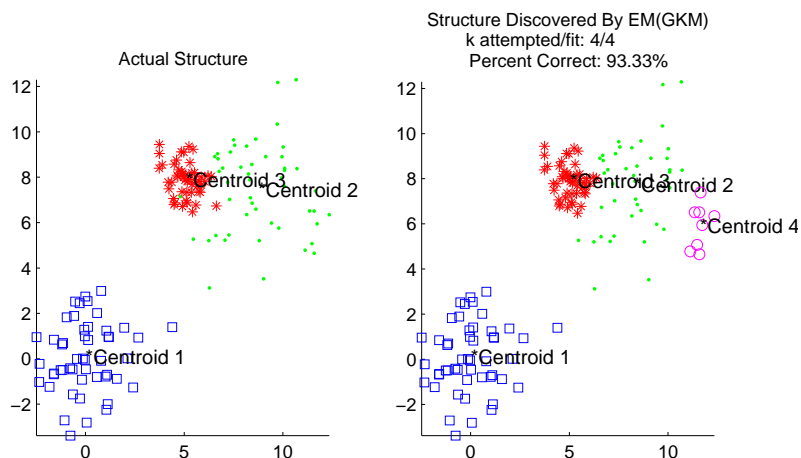


Figure 8.7: Simulation S3 - Scatter plot of  $\hat{K} = 4$  Model from EM(GKM).

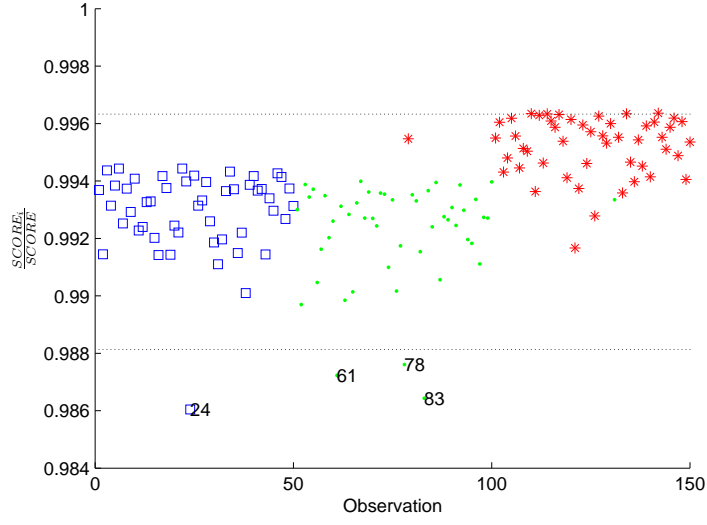


Figure 8.8: Simulation S3 - Influential Observation Detection Plot of  $\hat{K} = 3$  Model from EM(GKM).

Table 8.12: Simulation S3 - Actual and Estimated Parameters Using Best Model.

Actual Parameters			Estimated Parameters $\hat{K} = 4$		
$\pi_k$	$\mu_k$	$\Sigma_k$	$\hat{\pi}_k$	$\hat{\mu}_k$	$\hat{\Sigma}_k^*$
0.33	$\begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$	$\begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 2.00 \end{bmatrix}$	0.33	$\begin{bmatrix} 0.00 \\ 0.11 \end{bmatrix}$	$\begin{bmatrix} 1.85 & 0.27 \\ 0.27 & 2.25 \end{bmatrix}$
0.33	$\begin{bmatrix} 8.30 \\ 8.10 \end{bmatrix}$	$\begin{bmatrix} 4.00 & 0.00 \\ 0.00 & 4.00 \end{bmatrix}$	0.28	$\begin{bmatrix} 8.25 \\ 7.74 \end{bmatrix}$	$\begin{bmatrix} 1.91 & 1.02 \\ 1.02 & 3.87 \end{bmatrix}$
0.33	$\begin{bmatrix} 5.00 \\ 8.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 & 0.00 \\ 0.00 & 0.50 \end{bmatrix}$	0.33	$\begin{bmatrix} 5.04 \\ 7.91 \end{bmatrix}$	$\begin{bmatrix} 0.41 & -0.10 \\ -0.10 & 0.54 \end{bmatrix}$
-	-	-	0.053	$\begin{bmatrix} 11.61 \\ 5.90 \end{bmatrix}$	$\begin{bmatrix} 0.24 & 0.11 \\ 0.11 & 0.97 \end{bmatrix}$

**SUMMARY:** We fit the GMM using EM(GKM) to 150 observations, with the EM converging for all models in only one simulation. *AIC* and *ICOMP* picked a model with four groups misclassifying 10 observations; *SBC* and *ICOMP<sub>PEU</sub>* picked a model with three groups misclassifying only 2 observations. Several clearly separated observations were flagged as potential outliers.

#### 8.2.4 Real data - Iris

Now we show the application of these new methods to Fisher’s iris data. As mentioned previously, the trick is to have the data tell us that there are three groups, when all it knows is what’s shown in Figure 8.9. In all dimensions, measurements on the *Versicolor* and *Virginica* varieties overlap substantially, which seems to restrict the ability of the EM algorithm to converge as already shown. To determine the appropriate covariance regularization function for the regularized Mahalanobis distance, we fit  $K_{\max}$  mixtures to the data using each of the 4 robust covariance smoothers, and identified which was associated with the minimum of score for each IC. As shown in Table 8.13, the data suggests we should use the Convex Sum Estimator. The modeling results are summarized in Table 8.14. Note the similarity between the models selected by *SBC* and *ICOMP<sub>PEU</sub>*, with the consistent penalties. For this dataset, it is clear that the heavier penalty was not needed. Table 8.15 presents the results from the best model selected by both *AIC* and *ICOMP*. This model only misclassified  $\frac{5}{150} = 3.33\%$  of the observations; all 5 were really *Iris Versicolor*, but were classified as *Iris Virginica*. Since the confusion is so simple, we don’t show the confusion matrix. Given the amount of overlap exhibited by these groups, this performance is remarkable. Using *ICOMP* and the best mixture model it identified, we performed the complete enumerative subset analysis, with the results shown in Table 8.16. For each group of equally-sized subsets, the \* identifies the best. According to this table, we could fit the univariate mixture model to just the 2<sup>nd</sup> (*petal width*) or 4<sup>th</sup> (*sepal width*) variables and get a better mixture model, at least in terms of the IC score. Visual inspection of Figure A.14 in the appendix seems to confirm this. The next best subset model would use the {2,4} subset. In Figure A.14, this is the lower middle plot; again we see very good class separation here. Due to the overlap that remains in these three subset models, we would not expect to see any improvement in the accuracy of the mixture model, however. Further analysis confirmed this expectation. Visual inspection of the plots in Figure A.14 suggest there is at least one outlier value in the *Iris Setosa* group, and possibly two outliers in the *Iris Virginica* group. Influence detection identifies three observations that fall substantially short of the [0.970, 1.002]

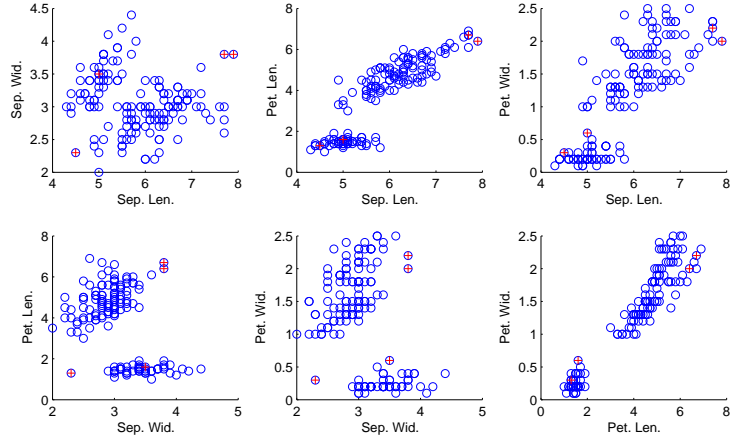


Figure 8.9: Iris data - Pairwise Scatter Plots.

Table 8.13: Iris data - Covariance Estimator Selection.

Smoother	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
$\Sigma_{MLE/EB}$	765.04	709.91	738.91	823.35
$\Sigma_{SRE}$	802.93	806.47	859.07	837.68
$\Sigma_{CSE}$	586.42*	554.81*	630.13*	717.43*
$\Sigma_{THOMAZ}$	745.53	980.01	728.38	860.59

Table 8.14: Iris data - Mixture of Gaussian Models Selected by GEM(GARM).

Output	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
$\hat{K}$ fit	3	2	3	2
Score	511.01	626.79	469.27	583.36
Correct Classification Rate	96.67%	66.67%	96.67%	66.67%

Table 8.15: Iris data - Results from Best Replication Using  $ICOMP$ .

$\hat{K}$ attempted	$\hat{K}$ fit	$ICOMP$	Correct Classification Rate
1	1	820.01	33.33
2	2	501.81	66.67
<b>3*</b>	<b>3</b>	<b>469.27</b>	<b>96.67</b>
4	3	469.27	96.67
5	5	525.87	80.00
6	5	545.05	73.33

interval for  $I_i$ , confirming the visual suggestion. The ratios (0.958, 0.956, 0.963) all suggest that the mixture model could be improved if they were removed. Figure 8.10 has the influence detection plot. These four observations are marked in Figure 8.9 with the red  $+$ . Of course, it is difficult to visually evaluate these as outliers in the pairwise plots, though it does seem that at least two are justifiably flagged.

Table 8.16: Iris data - Subset Analysis Using Best  $\hat{K} = 3$  Mixture Model.

Subset	<i>ICOMP</i>
{1, 2, 3, 4}	469.27*
{2, 3, 4}	373.88*
{1, 3, 4}	433.16
{1, 2, 4}	442.75
{1, 2, 3}	581.09
{3, 4}	307.37
{2, 4}	279.38*
{2, 3}	493.39
{1, 4}	412.03
{1, 3}	533.31
{1, 2}	469.88
{4}	212.96
{3}	411.63
{2}	177.91*
{1}	359.03



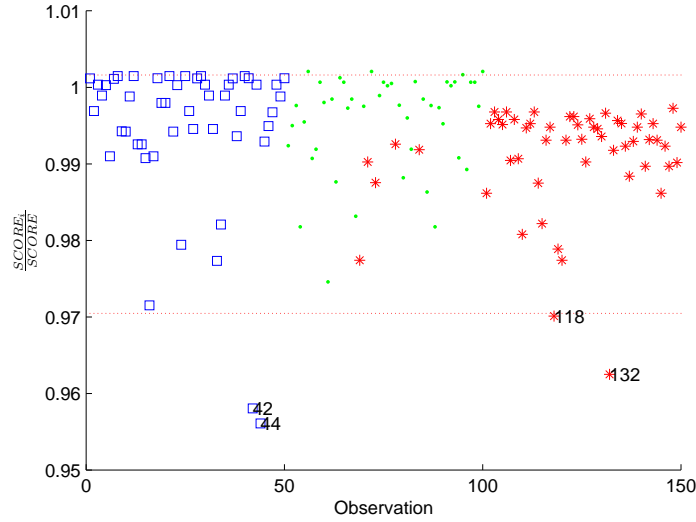


Figure 8.10: Iris data - Influence Detection Plot for  $\{2, 4\}$  Subset Model.

**SUMMARY:** We fit the GMM using GEM(GARM), with the Convex Sum covariance regularization. Both *AIC* and *ICOMP* identified a model with the correct structure, misclassifying only five (3.3%) observations. Using *ICOMP*, a subset model including only variables  $\{x_2, x_4\}$  was identified that did not allow for an improvement in classification error, but did allow for empirical identification of potential outliers.

### 8.2.5 Real data - Diabetic

Next we come back to the diabetic dataset reported in Andrews and Herzberg (1985). Five medical measurements relating to insulin usage were taken on  $n_1 = 33$  *overt diabetic*  $n_2 = 36$  *chemical diabetic*, and  $n_3 = 76$  *non-diabetic* patients. There is clear overlap among the groups, especially for  $x_4$  and  $x_5$ , and nice separation in the *Glucose Area* measurements, as can be seen in Figure A.12 in the appendix. Using the information criteria minimization for  $\hat{K} = 6$  to select the appropriate robust covariance estimator, we selected the Empirical Bayes estimator. We first evaluated this dataset using GEM initialized by GARM for the GMM. Table 8.17 shows the model selection results for the Gaussian mixture model. *ICOMP* provided the best performance - it only selected models with  $\hat{K} = 2$  or  $\hat{K} = 3$  groups. The simpler criteria, *AIC* and *SBC*, exhibited much less precision than either form of *ICOMP*. In Table 8.18, we show the top 5 models selected by *ICOMP*. The scores of the best 4 models are so close as to be indistinguishable.

Table 8.17: Diabetic data - Model Selection Frequencies out of 25 Replications.

$\hat{K}$	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
1	0	0	0	100*
<b>2</b>	<b>0</b>	<b>5</b>	<b>60*</b>	<b>0</b>
<b>3</b>	<b>0</b>	<b>20</b>	<b>40</b>	<b>0</b>
4	10	30*	0	0
5	45	35	0	0
6	45*	10	0	0

Table 8.18: Diabetic data - Summary of Best 5 Replications from the Gaussian Mixture Model.

$\hat{K}$	$ICOMP$	Correct Classification Rate
2	6418.82	71.03
2	6421.95	71.72
2	6423.28	73.10
2	6425.34	72.41
3	6452.23	86.21

**SUMMARY:** We fit the GMM using GEM(GARM), running 25 replications.  $AIC$  and  $SBC$  both overfit, while  $ICOMP_{PEU}$  selected  $\hat{K} = 1$  in all replications.  $ICOMP$  only chose models with  $\hat{K} = 2$  or  $\hat{K} = 3$ . The model with the best  $ICOMP$  score correctly classified 71% of the patients, though one of the top five models got 86.2% correct.

### 8.2.6 Real data - Cancer

Next, we revisit the breast cancer dataset already analyzed. Recall that there are  $n = 569$  observations,  $p = 30$  variables, and  $K = 2$  groups;  $n_1 = 212$  patients with *malignant* tumors and  $n_2 = 357$  with *benign* tumors. For a different way to look at the data, we used Multi-dimensional scaling to project the data into two and three dimensions, with the scatter plots shown in Figure 8.11. The two populations are identified with different markers. While they look linearly separable from this perspective, there is a clear appearance of a merged boundary. Next, we considered the distribution of the data using the tests for multivariate Gaussian skewness and kurtosis of Mardia (1974). For data following a multivariate Gaussian distribution, the theoretical population skewness and kurtosis parameters are  $\beta_1 = 0$  and  $\beta_2 = p(p + 2)$ . Mardia's sample values can be computed as in (8.1) and (8.3).

$$\hat{\beta}_1 = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \left[ (x_i - \bar{x}) \hat{\Sigma}^{-1} (x_j - \bar{x})' \right]^3 \quad (8.1)$$

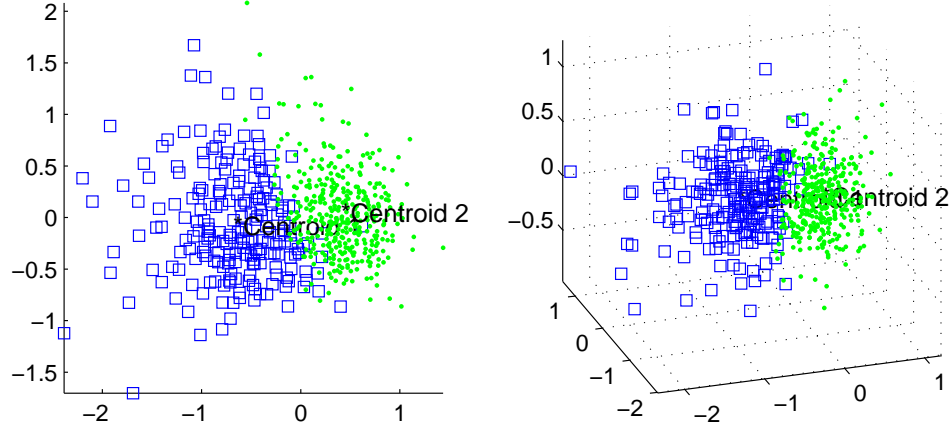


Figure 8.11: Cancer data - MDS Scatter Plots.

$$\chi^{2*} = \frac{n}{6} \hat{\beta}_1 \sim \chi^2 \left( \frac{p(p+1)(p+2)}{6} \right) \quad (8.2)$$

$$\hat{\beta}_2 = \frac{1}{n} \sum_{i=1}^n \left[ (x_i - \bar{x})' \hat{\Sigma}^{-1} (x_i - \bar{x}) \right]^2 \quad (8.3)$$

$$Z^* = \frac{(\hat{\beta}_2 - \beta_2)}{\sqrt{\frac{8p(p+2)}{n}}} \sim N(0, 1) \quad (8.4)$$

To test the null hypothesis  $H_0 : X \sim N_p(\mu, \Sigma)$  versus the alternative  $H_a : X \not\sim N_p(\mu, \Sigma)$ , we form the test statistics shown here; the test is one-sided for skewness, while the kurtosis test is two sided. For the this dataset, results clearly indicate nonnormal skewness and kurtosis, as shown in Table 8.19. The fact that the Gaussian mixture model will clearly be misspecified indicates the need for a robust criterion; the high dimensionality also indicates the need for a stronger penalty. As such, focus on  $ICOMP_{PEU\_MISP}$ . We fit the mixture of normals model for  $\hat{K} = 6$  to determine which covariance estimator should be used. Only the convex sum and Thomaz algorithms provided solutions. The others were unable to solve all numerical issues with the covariance matrix. Out of these two, the IC scores were all lower when  $\Sigma_{CSE}$  was utilized. With these determinations made, we used GARM to initialize GEM for the mixture of Gaussians model and executed 25 replications of the modeling process.  $AIC$  overshoot the  $\hat{K} = 2$  model in 68% of the runs while  $SBC$  did so in 52%. The regular  $ICOMP$  performed quite well - while the model it identified as best used  $\hat{K} = 3$  mixtures, it correctly classified the breast masses of  $\frac{519}{569} = 91.39\%$  of the patients. Only two observations were placed in the extraneous class. While this result is gratifying,  $ICOMP$  had a

difficult time choosing between two populations and three.  $ICOMP_{PEU\_MISP}$ , however, selected a model with two mixtures in all 25 replications. The best five models are shown in Table 8.20.

Note that the best model, as measured by the criterion, also had the best classification performance. In Table 8.21, we show the scores for  $\hat{K} = 1 \dots 6$  from this replication. It is interesting how the algorithms were able to realize the data was best represented by two groups, no matter how many were attempted. Table 8.22 has the confusion matrix; all misclassified tumors were *malignant* masses incorrectly identified as *benign*. From a diagnostic point of view, there is clearly room for improvement.

For this dataset, there are 1,073,741,823 possible nontrivial subsets. Using the mixture model identified by  $ICOMP_{PEU\_MISP}$ , we performed 10 runs of using the GA to do subset selection. Allowing it to run up to 60 generations with a population size of 30, the GA evaluated at most 18,000 *unique* subsets - a mere 0.0017% of the subset space. The top five subsets are shown in Table 8.23. Using this best subset, we then used GARM to initialize GEM for fitting the  $\hat{K} = 2$  mixture of Gaussians model, with 10 replications.  $ICOMP_{PEU\_MISP}$  identified two models that correctly classified over 93% of the patients - almost a 6% improvement in error. Table 8.24 shows the confusion matrix from this model. The false negatives (*malignant* masses incorrectly identified as *benign*) have dropped by a full 2/3, and the number of misclassified datapoints is half as was obtained when all  $p = 30$  variables were used! From this subset, we also see a dramatic improvement in the work required for classification. There is no need to measure the *radius*, *texture*, or *concave points*. Medical staff also need not worry about computing the mean or standard error of *smoothness*, nor anything but the mean of *concavity*. If any of these required a manual measurement process, this removes a lot of chance for error. Interestingly, according to the data source, the previous best classification results with this data, 97.5%, were obtained via an **exhaustive search** using a **supervised learning** procedure. Our results came close to this level of accuracy while neither requiring exhaustive search nor using the information about the true group structure. Finally, we used the subset model to identify influential observations, of which there were many, as can be seen in Figure 8.12. Note that since the  $ICOMP_{PEU\_MISP}$  scores are all negative, the observations above the line all indicate that they may be degrading the model performance. In fact, 9 of these 14 observations were misclassified in the original mixture model using all variables. These observations were flagged with no knowledge of the true group structure. Perhaps the procedure is discovering which points have a high probability of being misclassified?

Table 8.19: Cancer data - Normality Test Results.

Skewness		Kurtosis	
$\beta_1$	0	$\beta_2$	960
$\hat{\beta}_1$	938.65	$\hat{\beta}_2$	2370.06
$\chi^2*$	89015.53	$Z^*$	383.81
95% Region	[0, 5124.96]	95% Region	[-1.96, 1.96]
p-value	0.00000	p-value	0.00000
Conclusion	$X \approx N(\mu, \Sigma)$	Conclusion	$X \approx N(\mu, \Sigma)$

Table 8.20: Cancer data - Summary of Best 5 Replications from Mixture of Normals Model.

$\hat{K}$	$ICOMP_{PEU\_MISP}$	Correct Classification Rate
2*	126310.57	88.23
2	127928.78	74.34
2	128497.96	63.27
2	128574.15	63.45
2	128950.44	69.07

Table 8.21: Cancer data - Results from Best Replication.

$\hat{K}$ attempted(fit)	$ICOMP_{PEU\_MISP}$	Correct Classification Rate
1, 1	137741.77	62.74%
2, 2	129427.79	63.09%
3, 2*	126310.57	88.23%
4, 2	129427.79	63.09%
5, 2	129427.79	63.09%
6, 2	129427.79	63.09%

Table 8.22: Cancer data - Confusion Matrix from Best Mixture of Normals Model.

		Predicted		Total
		$k$		
Actual	1	145	67	<b>212</b>
	2	0	357	<b>357</b>
	Total	<b>145</b>	<b>424</b>	<b>569</b>

Table 8.23: Cancer data - Best Five Subset Models Chosen by  $ICOMP_{PEU}$ .

Subset	Score
$\{8, 9, 11, 15, 16, 17, 18, 19, 25, 26, 27, 28, 29\}$	-29447.318
$\{5, 9, 10, 16, 17, 19, 20, 27, 28, 30\}$	-28489.414
$\{5, 7, 8, 9, 10, 15, 18, 26, 27, 28\}$	-26158.904
$\{8, 9, 11, 15, 16, 17, 18, 19, 26, 27, 28\}$	-25433.283
$\{5, 6, 8, 9, 17, 19, 20, 26, 27, 29\}$	-24204.027

Table 8.24: Cancer data - Confusion Matrix from Best Subset GMM.

		Predicted		Total
		$k$		
Actual	1	191	21	<b>212</b>
	2	16	341	<b>357</b>
	Total	<b>207</b>	<b>362</b>	<b>569</b>

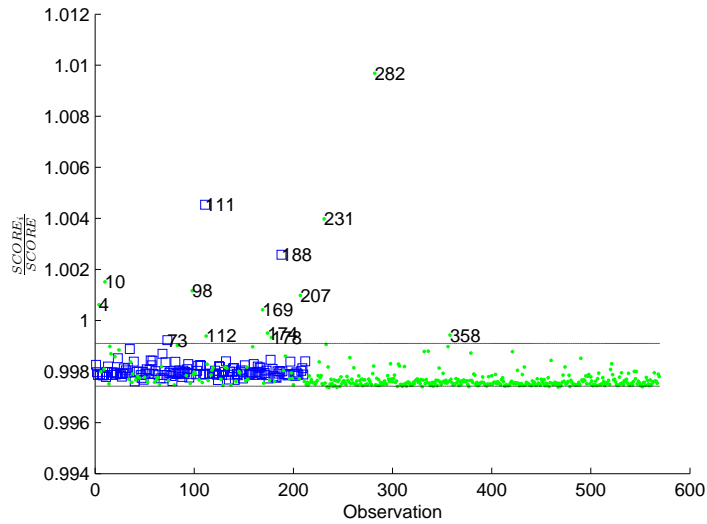


Figure 8.12: Cancer data - Detecting Influential Observations.

**SUMMARY:** We fit the GMM using GEM(GARM), running 25 replications with  $ICOMP_{PEU\_MISP}$ . The criterion identified a model with the correct structure of two populations, classifying 88% of the observations correctly.  $ICOMP$  was then used to identify a subset mixture model with 13 variables. GEM(GARM) was then run on the subset ten times, attempting to fit  $\hat{K} = 2$  populations to the data; two replications correctly classified 93% of the observations, in a more clinically useful manner. Several observations that had been misclassified in the original model were flagged as outliers.

### 8.2.7 Real data - Aorta

Our final example is with the aorta dataset already evaluated by the traditional GMM. The NMR aorta data analyzed here was collected by Pearlman (1986) at the Medical School of the University of Virginia. There are observations from  $n = 418$  patients on 16 different image acquisition variables. Including direction and orientation variables, we have  $p = 20$ . The first  $n_1 = 194$  patients exhibited *early atheroma*, and the remaining  $n_2 = 224$  patients were *healthy*. As can be seen in the appendix (Figures A.8 and A.9), the data exhibit distinct patterns of good and poor class separation, as well as marked non-normality. Preliminary analysis of this data suggested that the Empirical Bayes estimator was the best to use for this data. Using GEM initialized by GARM to fit a mixture of Gaussians clearly indicated the need for a strong misspecification-robust penalty.  $AIC$ ,  $SBC$ ,  $ICOMP$ , and  $ICOMP_{PEU}$  all performed rather poorly with this data - the first three tended to select  $\hat{K} = 4$ , and  $ICOMP_{PEU}$  selected  $\hat{K} = 1$ . We then ran 10 replications of the GMM using  $ICOMP_{PEU\_MISP}$  to drive model selection. In 100% of the runs, the criterion selected  $\hat{K} = 2$  mixtures. There was quite a bit of variation in correct classification rates, though, as can be seen in Table 8.25. Models selected in the three replications with the lowest scores, indicated by \*, correctly identified over 90% of the patients' heart tissue condition; the model with the minimum score only misclassified  $\frac{32}{418} = 7.66\%$  of the observations. The confusion matrix from this model is shown in Table 8.26. It is interesting to note that none of the patients exhibiting *early atheroma* (group 1) were mistakenly identified as being *healthy*. This is positive; in medical diagnosis, it's better to err on the side of caution and have some patients undergo further tests, than to have sick patients be classified as healthy. Though not clinically verified, it is reasonable to conjecture that the 32 misclassified patients were at some pre-arteriosclerosis stage. This would explain why their heart tissues were identified as being diseased. It would be interesting to have their records reevaluated, to determine if they developed atheroma of the aorta shortly after the imaging study.

Table 8.25: Aorta data - Models Selected by  $ICOMP_{PEU\_MISP}$  in All Replications.

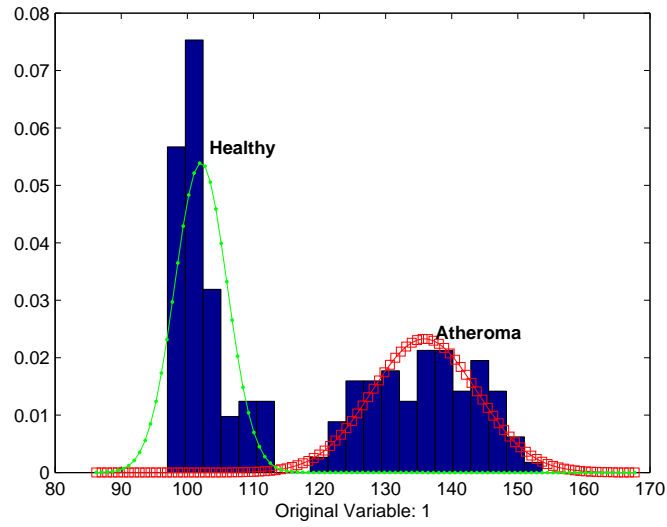
Replication	$\hat{K}$ fit	$ICOMP_{PEU\_MISP}$	Correct Classification Rate
1	2*	48104.91	92.34
2	2	50073.46	68.90
3	2	50470.33	63.88
4	2*	48828.91	90.91
5	2	52226.73	54.07
6	2*	48108.91	92.34
7	2	49011.94	85.65
8	2	50117.90	67.46
9	2	49011.94	85.65
10	2	49233.72	79.43

Table 8.26: Aorta data - Confusion Matrix from Best Gaussian Mixture Model.

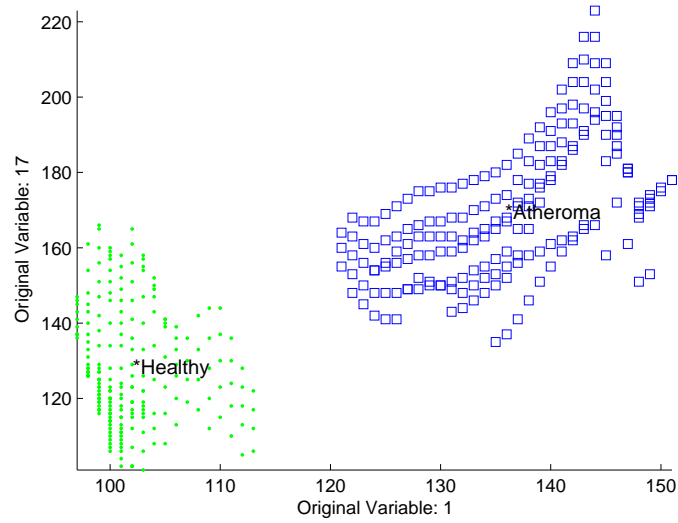
		Predicted		Total
		$k$		
Actual	1	194	0	<b>194</b>
	2	32	192	<b>224</b>
	Total	<b>226</b>	<b>192</b>	<b>418</b>

After identifying the mixture model that correctly classified 92.34% of the patients into  $\hat{K} = 2$  mixtures, we proceeded to use  $ICOMP_{PEU\_MISP}$  for the post subset and influence analysis. We executed 5 runs of the GA for subset modeling, which took less than a minute and a half. Table 8.27 shows some of the subsets identified by the GA. Thus, the process identified a single-variable model as needing only the 1<sup>st</sup> variable to give the optimum mixture model. Visual inspection of the 2-d scatter plot using the **estimated** class labels identified the  $\{1, 17\}$  subset as having very good estimated class separation. With these results, we fit a univariate and bivariate mixture model to the appropriate subsets. In both cases, this sequential process allowed us to increase our correct classification rate from 92.34% to 100%! The two plots in Figure 8.13 demonstrate the excellent separation between the two classes of patients. Finally, we considered the detection of influential variables using the  $\{1, 17\}$  subset model. With a baseline score of  $ICOMP_{PEU\_MISP} = 6842.025$ , the 95% interval for all IC ratios was  $[0.997, 0.998]$ . 10 observations were identified as falling below this range; and thus potentially influential observations. As can be seen in Figure 8.14, several of these observations are clearly separated. *Try to identify this visually using the entire dataset!*





(a) Best Univariate Subset Mixture Model.



(b) Best Bivariate Subset Mixture Model.

Figure 8.13: Aorta data - Subset Mixture Models with 0% Misclassification.

Table 8.27: Aorta data - Partial Post-subset Analysis Results Summary.

Subset	$ICOMP_{PEU\_MISP}$
$\{1\}$	3304.745*
$\{18\}$	3442.323
$\{3, 12\}$	5368.950*
$\{3, 18\}$	5514.481
$\{3, 19\}$	5925.296
$\{13, 18\}$	6200.085
$\{10, 12\}$	6398.004
$\{1, 17\}$	6842.025
$\{3, 11, 12\}$	7899.519*
$\{3, 9, 11\}$	8236.589

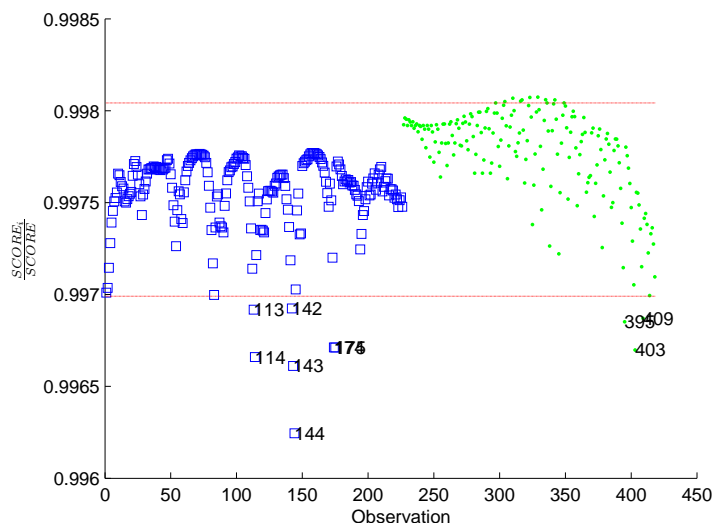


Figure 8.14: Aorta data - Influence Detection for Best Bivariate Subset Model.

**SUMMARY:** We fit the GMM using GEM(GARM), running 10 replications with  $ICOMP_{PEU\_MISP}$ . All runs resulted with the correct structure of  $\hat{K} = 2$  being selected, with the lowest score associated with a model correctly classifying 92.34% of the patients - with no false negatives. We then fit the identified mixture model to subsets of the variables using the GA, and identified several subsets which, when further analyzed, allowed for 100% classification. Using one of these subsets, 10 observations were flagged as being probable outliers.

### 8.3 ECMM

Here we apply the ECMM to three simulated and two real datasets. For the datasets evaluated, the Pearson Type VII model was picked as best most often. This result was unexpected, since this subclass doesn't adapt to peak behavior as well as the Kotz subclass. We also note the poorer performance of the *ICOMP* criteria with some of these datasets. It seems likely that this is due to how the IFIM was simplified to make the problem tractable. Finally, though our hybrid EM algorithm is an improvement in generality over the GMM EM, it is not a panacea for convergence issues. Here we take the opportunity to remind the reader that in tables recording model selection frequencies, we indicate the true structure with **bold** typeface, and the models selected by each criteria are indicated by an \* next to the score.

#### 8.3.1 Simulation S3 - Spherical Overlapping

For our first simulation study with  $n = 201$  (due to rounding) samples, we ran both the Kotz type and Pearson type VII mixture models on the simulation which featured spherical clusters with one group wholly contained in another. Information criteria scores indicated that the Pearson type VII model was more appropriate for this dataset, as shown in Table 8.28, and it also resulted in typically lower error rates. Model selection frequencies out of  $M = 50$  simulations are shown in Table 8.29. The first three criteria formulated for the Pearson type VII mixture model picked the correct structure most often. As with the Gaussian mixture model, the simpler structure in which the hidden group was not identified was never selected. Table 8.30 shows the model scores and classification rates for  $\hat{K} = 1 \dots 6$  from the Pearson type VII simulation with the lowest *ICOMP* score. Note how the algorithm identified the correct structure even when fitting  $\hat{K} = 4, 5, 6$  groups to the simulated data. Table 8.31 shows the confusion matrices from the Kotz type and Pearson type VII mixture models, as identified by *ICOMP*. The best Pearson type VII model used shape parameters  $\hat{\nu}_1 = 40.40$  and  $\hat{\nu}_2 = \hat{\nu}_3 = 4.00$ ; the best Kotz type model came up with  $\hat{\beta}_1 = 1.07$ ,

Table 8.28: Simulation S3 - ECMM Subclass Selection Results.

IC	Kotz	Pearson Type VII
<i>AIC</i>	1889.61	1783.76*
<i>SBC</i>	2081.62	1832.44*
<i>ICOMP</i>	1659.71	1519.88*
<i>ICOMP<sub>PEU</sub></i>	1338.15	1183.60*

Table 8.29: Simulation S3 - Model Selection Frequencies using ECMM with GEM(GARM).

$\hat{K}$	Kotz Type				Pearson Type VII			
	<i>AIC</i>	<i>SBC</i>	<i>ICOMP</i>	<i>ICOMP</i> <sub>PEU</sub>	<i>AIC</i>	<i>SBC</i>	<i>ICOMP</i>	<i>ICOMP</i> <sub>PEU</sub>
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
<b>3</b>	<b>36*</b>	<b>60*</b>	<b>54*</b>	<b>10</b>	<b>96*</b>	<b>100*</b>	<b>96*</b>	<b>14</b>
4	40	34	40	58	4	0	4	36
5	22	6	4	16*	0	0	0	30*
6	2	0	2	16	0	0	0	20

Table 8.30: Simulation S3 - Model Scores from Best Pearson type VII Mixture Model Selected by *ICOMP*.

$\hat{K}$ attempted,fit	<i>ICOMP</i>	Correct Classification Rate
1, 1	2116.69	33.33
2, 2	1592.50	66.67
<b>3, 3*</b>	<b>1425.90</b>	<b>98.01</b>
4, 3	1425.90	98.01
5, 3	1425.90	98.01
6, 3	1429.04	96.52

Table 8.31: Simulation S3 - Confusion Matrices for Best Kotz Type and Pearson Type VII Mixture Models.

	<i>PVII</i>	Predicted			Total
	<i>k</i>	1	2	3	
Actual	1	67	0	0	<b>67</b>
	2	0	63	4	<b>67</b>
	3	0	0	67	<b>67</b>
	Total	<b>67</b>	<b>63</b>	<b>71</b>	<b>201</b>

	<i>KT</i>	Predicted			Total
	<i>k</i>	1	2	3	
Actual	1	67	0	0	<b>67</b>
	2	0	52	15	<b>67</b>
	3	0	1	66	<b>67</b>
	Total	<b>67</b>	<b>53</b>	<b>81</b>	<b>201</b>

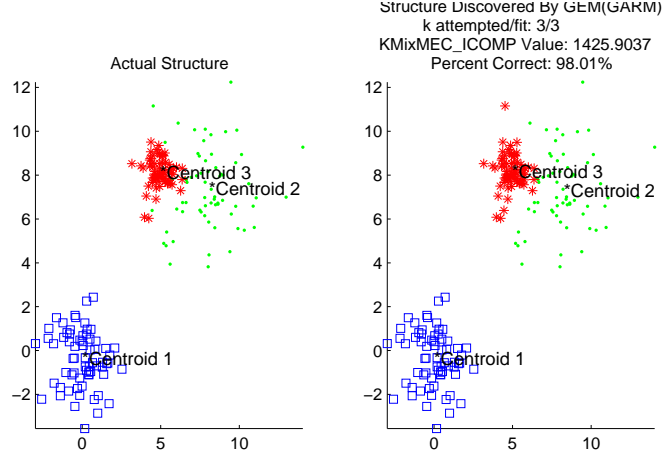


Figure 8.15: Simulation S3 - Actual Grouping Structure and Structure Estimated by Pearson Type VII Mixture Model.

$\hat{\beta}_2 = 0.99$ , and  $\hat{\beta}_3 = 1.18$ . From the first model, we see slightly heavier tails in the two overlapped groups, and in the second model, we see heavier tails and a higher peak in only one of the groups. The variance estimates differed from the true variances in the manner that would be expected - increase the tails and decrease the variance. In Figure 8.15, we show the actual and estimated structure from the best model which only misclassified 4 datapoints.

**SUMMARY:** IC scores identified the Pearson Type VII model and the empirical Bayes covariance estimator as more appropriate for this dataset. We ran fifty simulations with the ECMM for both EC mixture models, with the Pearson Type VII model picking the correct structure with high frequency. The best Pearson Type VII model identified by *ICOMP* only misclassified four of the 201 samples. The two overlapping populations were both modeled with heavy tails:  $\hat{\nu} = 4.00$ . Using the EM initialized by K-Means for the GMM, the likelihood selected a group structure with a single population; EM rarely converged. Still with convergence issues, the information criteria using EM(GKM) for the Gaussian mixture model either selected the true model with 99% correct classification, or a model with an extra population (93%). With this dataset, the classification rate was similar, but we gained in generality and lost the convergence issues.

### 8.3.2 Simulation S2 - Ellipsoidal Overlapping

The second simulated dataset for which results are reported here is that already analyzed by the GMM in Section 8.2.2. Our first step is to determine which EC subclass fits the data better, as a

Table 8.32: Simulation S2 - EC Subclass Selection Results.

Criteria	Kotz Score	Pearson VII Score
<i>AIC</i>	1314.07	1026.78*
<i>SBC</i>	1186.66	1075.91*
<i>ICOMP</i>	962.76	813.09*
<i>ICOMP<sub>PEU</sub></i>	250.34*	539.31

Table 8.33: Simulation S2 - Pearson type VII Mixture Model Selection Frequencies.

$\hat{K}$	<i>AIC</i>	<i>SBC</i>	<i>ICOMP</i>	<i>ICOMP<sub>PEU</sub></i>
1	0	0	0	0
2	0	0	0	0
<b>3</b>	<b>76*</b>	<b>90*</b>	<b>52</b>	<b>0</b>
4	22	10	30*	16
5	2	0	16	58*
6	0	0	0	26
Best	93.17%	93.17%	67.17%	68.67%

mixture model. Table 8.32 shows information criteria scores for both the Kotz type and Pearson type VII subclass, when used in a mixture model with  $\hat{K} = 6$  groups. Using a majority voting rule, we would determine that we should fit a mixture of Pearson type VII distributions to this simulation. Subsequently, we ran  $M = 50$  simulations from this protocol with  $n = 600$  observations, fitting Pearson type VII ECMs. Table 8.33 summarizes the results. It is interesting that, for this example, neither of the *ICOMP* criteria performed as well as *AIC* and *SBC*. *SBC* both picked the correct structure as the best model overall and picked it with the highest frequency. The same simulation resulted in the overall best for both *AIC* and *SBC*; in fact, GEM converged to the same solution for each. In Tables 8.34 and 8.35, we show the estimated parameters from the best model identified by *SBC*, along with the confusion matrix showing only 41 misclassified observations. A quick foray back into the appendix shows us that the estimated parameters are actually very similar to those actually used to generate the data - especially for the covariance matrix. Even though the data were generated from a multivariate Gaussian distribution, a high amount of accuracy was obtained by using slightly heavier tails - note the low values for the shape (degrees of freedom) parameter. Finally, we can see in Figure 8.16 how quickly GEM found the final solution - it was identified by the 15<sup>th</sup> generation.

Table 8.34: Simulation S2 - Pearson type VII Mixture Model Parameter Estimates.

Group	$\hat{\pi}$	$\hat{\mu}$	$\hat{\Sigma}_{PVI}$	$\hat{\nu}$
1	25.67	[ 0.55 0.93 ]	$\begin{bmatrix} 1.43 & 0.44 \\ 0.44 & 0.16 \end{bmatrix}$	7.05
2	54.50	[ 0.94 0.65 ]	$\begin{bmatrix} 0.30 & -0.21 \\ -0.21 & 0.16 \end{bmatrix}$	4.23
3	19.83	[ 0.34 -0.49 ]	$\begin{bmatrix} 0.15 & 0.06 \\ 0.06 & 0.04 \end{bmatrix}$	4.54

Table 8.35: Simulation S2 - Confusion Matrix from Best ECMM.

		Predicted			Total
		$k$	1	2	
Actual	1	147	33	0	<b>180</b>
	2	7	293	0	<b>300</b>
	3	0	1	119	<b>120</b>
	Total	<b>154</b>	<b>327</b>	<b>119</b>	<b>600</b>

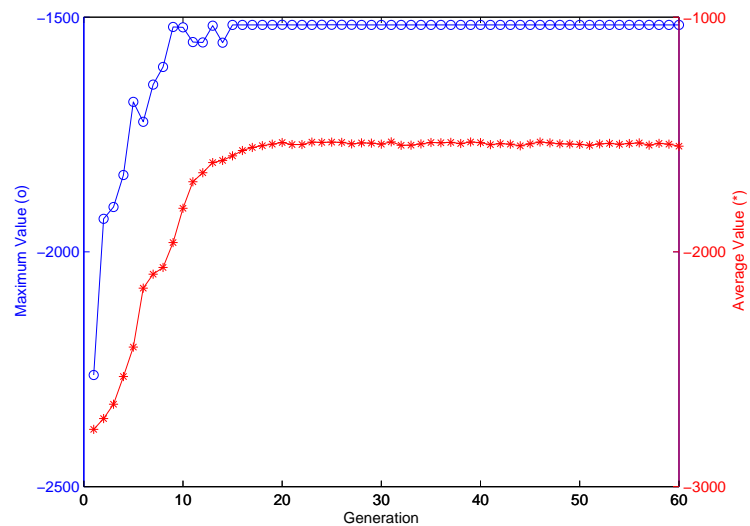


Figure 8.16: Simulation S2 - Progress Plot for GEM Showing Quick Solution Identification.

**SUMMARY:** IC scores identified the Pearson Type VII model as better than the Kotz type distribution for  $n = 600$  observations from this dataset. *SBC* identified a model with the correct structure in 90% of the simulations, and was minimized with a model correctly identifying 93% of the observations into three populations. The shape parameters were estimated a  $\hat{\nu}_1 = 7.0$ ,  $\hat{\nu}_2 = 4.2$ , and  $\hat{\nu}_1 = 4.5$  - heavier than Gaussian. The EM for the Gaussian mixture model was unable to converge consistently, but in two simulations, it converged for  $\hat{K} = 1 \dots 4$ . *ICOMP* was minimized for a model with three groups, for a slightly better classification rate of 95%.

### 8.3.3 Simulation S5 - Mixed Overlapping

This simulation was designed purposefully with the Kotz distribution in mind. Recall that the PE distribution is a special case of the Kotz subclass. This simulation protocol generates three groups from the PE with different shapes - bell-curved, heavily peaked, and flat. One of the groups is generated so as it could overlap both the others. We would hope that fitting a mixture of Kotz type distributions would allow us to model each of these shapes simultaneously. We fit  $\hat{K} = 1 \dots 6$  to  $M = 100$  simulations with  $n = 300$  observations. Using GEM initialized by GARM, *AIC* only picked the true structure of  $K = 3$  groups in eight simulations; all criteria either picked the true structure, or overfit. *SBC* selected three groups 17% of the time, and *ICOMP* did so at 20%. Finally, *ICOMP<sub>PEU</sub>* had the worst performance, picking the correct structure in only six simulations. Table 8.36 shows the top five models selected by each criteria - the first number is in each row  $\hat{K}$ , and the second is the percent of observations correctly classified. At the bottom, we also indicate the IC scores for the best two models. It is interesting to note the high accuracy of some of the models, even though all the top five models had at least one extra group. In Table 8.37,

Table 8.36: Simulation S5 - Top Five EC Mixture Models from GEM(GARM) by Criteria.

<i>AIC</i>		<i>SBC</i>		<i>ICOMP</i>		<i>ICOMP<sub>PEU</sub></i>	
4	86.00%	4	84.67%	4	90.00%	5	568.00%
4	87.33%	3	94.33%	4	90.33%	5	82.33%
4	94.67%	4	92.67%	5	69.67%	6	81.00%
4	84.67%	4	94.67%	3	93.67%	6	76.00%
5	84.67%	4	92.33%	4	83.67%	5	66.67%
2271.99		2385.42		1739.38		1553.29	
2309.11		2396.38		1809.77		1587.17	



Table 8.37: Simulation S5 - Confusion Matrices from Best EC Mixture Models Identified by *SBC* and *ICOMP*.

Actual	<i>SBC</i>	Predicted				Total
	$k$	1	2	3	4	
	1	46	2	4	38	<b>90</b>
	2	0	120	0	0	<b>120</b>
	3	2	0	88	0	<b>90</b>
	Total	48	124	92	38	<b>300</b>

Actual	<i>ICOMP</i>	Predicted				Total
	$k$	1	2	3	4	
	1	61	2	7	20	<b>90</b>
	2	0	120	0	0	<b>120</b>
	3	0	0	89	1	<b>90</b>
	Total	61	122	96	21	<b>300</b>

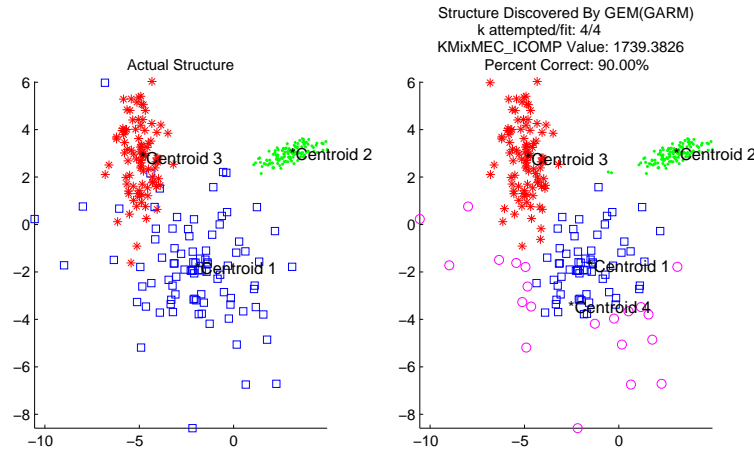


Figure 8.17: Simulation S5 - Actual and Estimated Grouping Structure.

we have the confusion matrices from the best models selected by *SBC* and *ICOMP* - with one extra group in each. The first model placed 46 observations in incorrect groups, while the second only missed 30. Finally, Figure 8.17 shows the actual and estimated group structure identified as the overall best by *ICOMP*. Datapoints from the first population which surrounded the rest of the other data at a distance were placed in their own group, centered relatively close to the overall center.

**SUMMARY:** We fit the Kotz mixture model to 100 simulations from this protocol using 300 observations. The top two models identified by *ICOMP* fit a spurious population, but correctly classified at least 90% of the samples. Regarding selection frequencies *SBC* and *ICOMP* performed the best, selecting the true structure of three groups in 17 and 20 of the simulations, respectively.

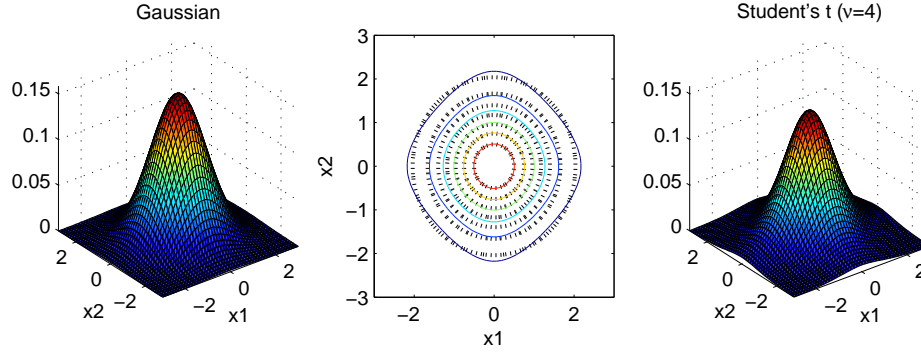


Figure 8.18: Demonstrating Slightly Heavier Tails for Bivariate PVII with  $\nu = 4$ .

### 8.3.4 Real data - Iris

For the iris dataset, we applied the EC mixture model, using GARM to initialize the hybrid EM algorithm. When the Kotz subclass was used, the algorithm was unable to converge except for the  $\hat{K} = 4$  model. Using the Pearson type VII, however, the EM algorithm converged for  $\hat{K} = 1 \dots 4$  - enough to identify a turning point for all information criteria scores.  $SBC$  and  $ICOMP$  were both minimized for the model with two groups for a correct classification rate of 66.67%.  $AIC$  and  $ICOMP_{PEU}$ , however, both homed in on the correct structure, and only misclassified  $\frac{7}{150} = 95.33\%$  of the observations. As noted before, the confusion was between the *Versicolor* and *Virginica* species. The confusion matrix from this model is shown in Table 8.38. The best model used a shape parameter of  $\nu_k = 4$  for all three groups. To visualize the slightly different tail behavior this imparts, Figure 8.18 shows the bivariate Gaussian pdf surface (left pane). In the right pane, we have the Pearson type VII (reduced to Student's t) density surface computed for four degrees of freedom. The center plot contrasts the contours, with the dashed black lines coming from the Gaussian density.

Table 8.38: Iris data - Confusion Matrix from Best Pearson Type VII Mixture Model.

		Predicted			
Actual	$k$	1	2	3	Total
	1	50	0	0	<b>50</b>
	2	0	48	2	<b>50</b>
	3	0	5	45	<b>50</b>
	Total	<b>50</b>	<b>45</b>	<b>55</b>	<b>150</b>

**SUMMARY:** We used the hybrid EM algorithm initialized by GARM for the Pearson type VII ECMM for the iris data. *SBC* and *ICOMP* picked models with only two populations, confounding *Versicolor* and *Virginica*. *AIC* and *ICOMP*, however, picked the correct structure and only misclassified seven of the 150 flowers. For the Gaussian mixture model, using GEM initialized by GARM, *AIC* and *ICOMP* were minimized for  $\hat{K} = 3$ , with an error of five flowers. When EM was used to fit the GMM, however, it was only able to converge when fitting two or three groups.

### 8.3.5 Real data - Diabetic

Finally, we have results from GEM(GARM) on the diabetic dataset. As with the Gaussian mixture model, we use the MLE/EB covariance estimator. Preliminary analysis seemed to give no preference to either the Kotz type or Pearson type VII distribution, so here we report results from fitting mixtures of Kotz's distributions. In Figure 8.19, we have the bivariate and trivariate scatter plots for the MDS dimensionally-reduced data. These plots bolster the claim of similarity between the *chemical diabetic* and *non-diabetic* patients. We performed  $M = 25$  replications of the modeling process, with model selection frequencies shown in Table 8.39. Keeping in mind that both a model with two groups or a model with three groups is considered “correct”, both *ICOMP*s did very well - selecting  $\hat{K} = 2, 3$  in 100% and 96% of the replications, respectively. Using *ICOMP*<sub>PEU</sub>, we show the scores and classification rates for the best five models in Table 8.40.

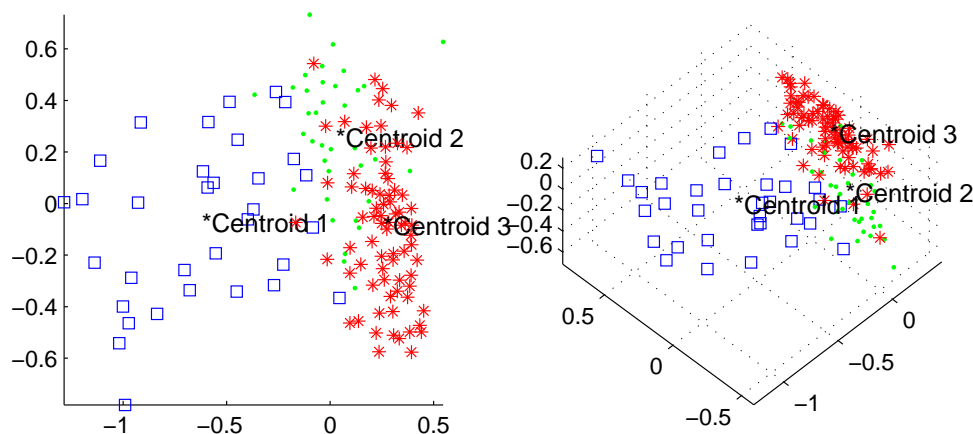


Figure 8.19: Diabetic data - MDS Scatter Plots.

Correctly classifying all but 12 of the patients, the EC mixture model performed the best out of all three mixture models fit to the diabetic data. Recall that, with the GMM, a misclassification rate of approximately 14% was achieved for this dataset, though this was for the 5<sup>th</sup> lowest score. The shape parameters identified by this model only indicated a slight departure from normality:  $\hat{\beta}_1 = 0.98$ ,  $\hat{\beta}_2 = 0.97$ , and  $\hat{\beta}_3 = 1.02$ . Finally, Table 8.41 has the confusion matrix from this model. Note that most of the confusion - eight of the 12 misclassified patients - was between the *chemical diabetic* and *non-diabetic* groups. Four of the patients were traded between *overt diabetic* and *chemical diabetic*. These small “trading” errors are not surprising, given the gradual manner in which this disease progresses.

Table 8.39: Diabetic data - Kotz type Mixture Model Selection Frequencies.

$\hat{K}$	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
1	0	0	0	0
<b>2</b>	<b>0</b>	<b>0</b>	<b>88*</b>	<b>60</b>
<b>3</b>	<b>0</b>	<b>8</b>	<b>12</b>	<b>36*</b>
4	16	40	0	0
5	44	28	0	0
6	40*	24*	0	4
Best	82.76%	84.14%	74.48%	91.72%

Table 8.40: Diabetic data - Best Five Kotz ECMMs Determined by  $ICOMP_{PEU}$ .

$\hat{K}$	$ICOMP_{PEU}$ Score	Correct Classification Rate
3	6265.35	91.72%
3	6265.99	91.03%
3	6271.67	91.03%
3	6280.74	89.65%
2	6287.82	74.48%

Table 8.41: Diabetic data - Confusion Matrix from Best EC Mixture Model.

		Predicted				
		$k$	1	2	3	Total
Actual	1	31	2	0		<b>33</b>
	2	2	31	3		<b>36</b>
	3	0	5	71		<b>76</b>
Total		<b>33</b>	<b>38</b>	<b>74</b>		<b>145</b>

**SUMMARY:** With no preference for either ECMM exhibited by the information criteria, we fit the mixtures of Kotz distributions to the diabetic data.  $AIC$  and  $SBC$  overfit very heavily, while  $ICOMP$  and  $ICOMP_{PEU}$  fit either of the correct models in 100% and 96% of the 25 replications. While only using slight departures from normality, the best model selected by  $ICOMP_{PEU}$  correctly classified 92% of the patients. The misclassifications could be characterized as small trading errors, which would be expected given the nature of the disease. Fitting the traditional GMM, we would choose a model with four groups, classifying 77% of the patients correctly. The updated Gaussian mixture model fared better at picking the correct structure, with  $ICOMP$  only choosing either of the correct models. The lowest  $ICOMP$  score was for a model with a classification rate of 71%.

## 8.4 KMM

In this section, we show results from fitting the mixture of kernel density estimators to simulated and real data. We tended to use fewer simulations or replication in this section, due to the higher computational burden associated with kernel density estimation.

### 8.4.1 Simulation S2 - Ellipsoidal Overlapping

We now revisit the dataset analyzed by both mixture models based on distributional assumptions. Stepping past the issues of non-convergence, we performed  $M = 20$  Monte-Carlo simulations (with  $n = 150$ ), fitting the mixture of kernel density estimators model. For these simulations, the bandwidth matrices were estimated using  $H_k = \frac{1}{n}W_k$ . Initialization was performed by GARM, and GEM was used for optimization. Table 8.42 demonstrates how well all four information criteria performed. All four criteria honed in on the correct structure of  $K = 3$  in at least 90% of the simulations. Using *ICOMP*,  $M = 18$  of the simulations selected a model with  $K = 3$  mixtures; a 95% confidence interval of the correct classification rate is given by [87.22%, 99.97%]. The minimum score across all simulations produced a model in which only 12 datapoints were misclassified, leading to a correct classification rate of  $1 - \frac{12}{150} = 92\%$ . However, looking across all simulations, two produced a model with a 96.67% correct classification rate. Results from one of these simulations are shown in Tables 8.43 through 8.45. The estimated parameters are computed based on the estimated class labels. In this best model, only 5 datapoints were misclassified. Figure 8.20 displays the scatter plots of the data with the true (left pane) and estimated (right pane) labels. The datapoints that were placed in the wrong mixture are identified with the black diamonds with black centers. It is clear from visual inspection of this plot that these observations would seem appropriate in either cluster 1 or cluster 2.

Table 8.42: Simulation S2 - Model Selection Frequencies for the Kernel Mixture Model.

$\hat{K}$	<i>AIC</i>	<i>SBC</i>	<i>ICOMP</i>	<i>ICOMP</i> <sub>PEU</sub>
1	0	0	0	0
2	0	0	0	0
<b>3</b>	<b>90*</b>	<b>95*</b>	<b>90*</b>	<b>95*</b>
4	10	0	5	0
5	0	5	5	5
6	0	0	0	0

Table 8.43: Simulation S2 - Results from Best Simulation Using the Kernel Mixture Model.

$\hat{K}$ attempted	$\hat{K}$ fit	ICOMP	Correct Classification Rate
1	1	673.42	50.00
2	2	630.41	68.00
<b>3*</b>	<b>3</b>	<b>536.88</b>	<b>96.67</b>
4	4	561.46	84.00
5	3	536.88	96.67
6	6	601.48	76.67

Table 8.44: Simulation S2 - Confusion Matrix from Best Simulation Using the KMM.

Actual Parameters			Estimated Parameters		
$\pi_k$	$\mu_k$	$\Sigma_k$	$\hat{\pi}_k$	$\hat{\mu}_k$	$\hat{\Sigma}_k^*$
0.3	$\begin{bmatrix} 0.6 \\ 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.2 & 0.3 \\ 0.3 & 0.1 \end{bmatrix}$	0.3	$\begin{bmatrix} 0.6 \\ 1.0 \end{bmatrix}$	$\begin{bmatrix} 1.0 & 0.3 \\ 0.3 & 0.2 \end{bmatrix}$
0.5	$\begin{bmatrix} 1.0 \\ 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.3 & -0.2 \\ -0.2 & 0.2 \end{bmatrix}$	0.5	$\begin{bmatrix} 1.0 \\ 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.4 & -0.2 \\ -0.2 & 0.2 \end{bmatrix}$
0.2	$\begin{bmatrix} 0.3 \\ -0.5 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.1 \\ 0.1 & 0.03 \end{bmatrix}$	0.2	$\begin{bmatrix} 0.3 \\ -0.5 \end{bmatrix}$	$\begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}$

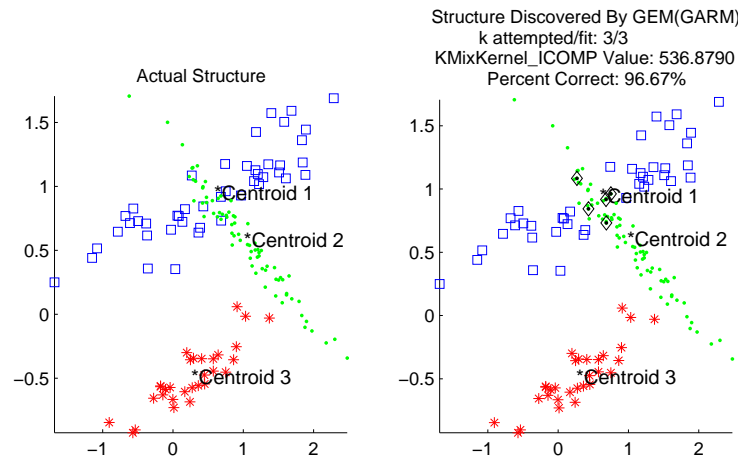


Figure 8.20: Simulation S2 - Scatter Plot of Best Model Using the Mixture of Kernels.

Table 8.45: Simulation S2 - Parameter Estimates from Best Simulation Using the Kernel Mixture Model.

Estimated Parameters			Actual Parameters		
$\hat{\pi}_k$	$\hat{\mu}_k$	$\hat{\Sigma}_k^*$	$\pi_k$	$\mu_k$	$\Sigma_k$
0.27	$\begin{bmatrix} 0.59 \\ 0.95 \end{bmatrix}$	$\begin{bmatrix} 1.04 & 0.32 \\ 0.32 & 0.15 \end{bmatrix}$	0.30	$\begin{bmatrix} 0.63 \\ 0.97 \end{bmatrix}$	$\begin{bmatrix} 1.17 & 0.33 \\ 0.33 & 0.12 \end{bmatrix}$
0.53	$\begin{bmatrix} 0.98 \\ 0.61 \end{bmatrix}$	$\begin{bmatrix} 0.35 & -0.22 \\ -0.22 & 0.18 \end{bmatrix}$	0.50	$\begin{bmatrix} 1.01 \\ 0.59 \end{bmatrix}$	$\begin{bmatrix} 0.34 & -0.24 \\ -0.24 & 0.18 \end{bmatrix}$
0.20	$\begin{bmatrix} 0.26 \\ -0.49 \end{bmatrix}$	$\begin{bmatrix} 0.36 & 0.10 \\ 0.10 & 0.16 \end{bmatrix}$	0.20	$\begin{bmatrix} 0.32 \\ -0.51 \end{bmatrix}$	$\begin{bmatrix} 0.14 & 0.05 \\ 0.05 & 0.03 \end{bmatrix}$

**SUMMARY:** Out of 20 simulations, all four criteria selected the true structure with a frequency of at least 90%. The minimum *ICOMP* score was associated with a model using  $\hat{K} = 3$  groups, misclassifying 12 observations; in two simulations, *ICOMP* selected a model with the correct structure correctly classifying 97% of the observations. The recovered mean vectors and covariance matrices from this model were very close to the actual values. The EM for the Gaussian mixture model was unable to converge consistently, but in two simulations, it converged for  $\hat{K} = 1 \dots 4$ . *ICOMP* was minimized for a model with three groups, for a classification rate of 95%.

#### 8.4.2 Real data - Diabetic

After fitting both the GMM and ECMM to the diabetic data, we evaluated this dataset using GEM initialized by GARM for the mixture of kernels model. We used the general bandwidth estimator (method 3 in Table 6.2). *ICOMP* provided the best performance, as shown in Table 8.46 - it only selected models with  $\hat{K} = 2$  or  $\hat{K} = 3$  groups. The simpler criteria, *AIC* and *SBC*, exhibited much less precision than either form of *ICOMP*. It is not surprising that *ICOMP*<sub>PEU</sub> selected the simpler model in which the clinically similar patients are grouped together. In Table 8.47, we show the top 5 models selected by *ICOMP*. The scores of the best 3 models are so close as to be indistinguishable, so we report the summary and confusion matrix from the replication with the highest correct classification rate, 85.5%, in Tables 8.48 and 8.49. In the confusion matrix, we



Table 8.46: Diabetic data - Model Selection Frequencies out of Five Replications.

$\hat{K}$	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
1	0	0	0	0
<b>2</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>100*</b>
<b>3</b>	<b>20*</b>	<b>80*</b>	<b>100</b>	0
4	80*	20	0	0
5	0	0	0	0
6	0	0	0	0

Table 8.47: Diabetic data - Summary of All Replications from the Kernel Mixture Model.

$\hat{K}$	$ICOMP$	Correct Classification Rate
3	6217.18	84.83
3	6218.08	85.52
3	6218.81	84.14
3	6230.86	84.83
3	6254.66	82.76

Table 8.48: Diabetic data - Results from Best Mixture of Kernels Replication.

$\hat{K}$ attempted	$\hat{K}$ fit	$ICOMP$	Correct Classification Rate
1	1	6532.92	52.41
2	2	6319.96	71.72
<b>3</b>	<b>3*</b>	<b>6218.08</b>	<b>85.52</b>
4	3	6244.14	71.38
5	4	6297.05	68.97
6	4	6339.67	74.48

Table 8.49: Diabetic data - Confusion Matrix from Best Replication.

		Predicted				
		$k$	1	2	3	Total
Actual	1	26	7	0	<b>33</b>	
	2	0	26	10	<b>36</b>	
	3	0	4	72	<b>76</b>	
	Total	<b>26</b>	<b>37</b>	<b>82</b>	<b>145</b>	

see that two thirds of the misclassified observations were either *chemical diabetic* or *non-diabetic* patients. It would be interesting to do a follow-up study to determine if any of these patients progressed to overt diabetes shortly after this data was gathered. We finish up this example with the best 10 subsets; fitting the mixture of kernels model to all 31 nontrivial subsets of the original variables required almost exactly 10 seconds. The results are shown in Table 8.50. All models listed here include either *Relative Weight* or *Fasting Plasma Glucose*, both of which are known to be important to diabetes. Based on the best subset identified,  $\{1, 2\}$ , we then evaluated the influence of all  $n = 145$  datapoints. Four were identified as being potentially influential, as shown in Figure 8.21. None of the influential observations were misclassified - all were patients in the *overt diabetic* group. Finally, we fit the KMM to the subset, but observed no improvement in classification accuracy.

Table 8.50: Diabetic data - Best 10 Subsets from Mixture of Three Kernel Density Estimators Model.

Subset	<i>ICOMP</i>	Subset	<i>ICOMP</i>
$\{1, 2\}$	1284.07	$\{2, 5\}$	2972.18
$\{1, 5\}$	1640.46	$\{1, 2, 3\}$	2988.40
$\{1, 4\}$	1690.87	$\{2, 3\}$	3000.16
$\{1, 3\}$	1825.87	$\{1, 2, 4\}$	3000.44
$\{1, 2, 5\}$	2899.39	$\{2, 4\}$	3017.10

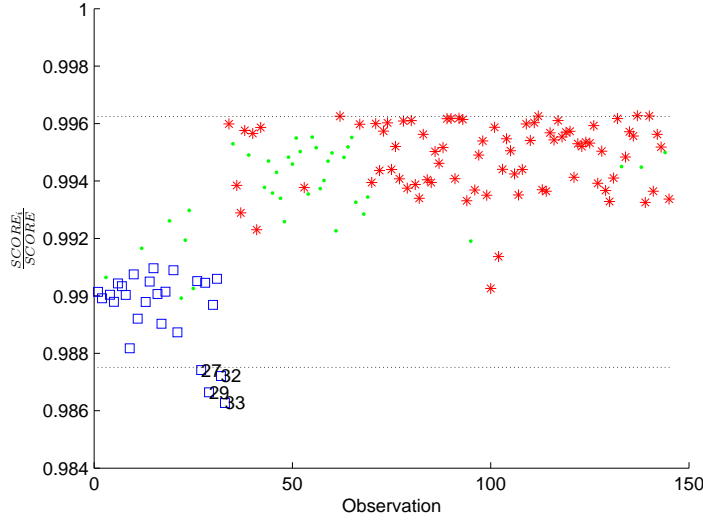


Figure 8.21: Diabetic data - Detecting Influential Observations.

**SUMMARY:** The best model chosen by *ICOMP* out of five replications correctly classified 85% of the patients into three populations. *ICOMP* settled on  $\hat{K} = 3$  in all replications; *ICOMP*<sub>PEU</sub> chose  $\hat{K} = 2$  in all. Using the best *ICOMP* model, we were able to reduce the dimensionality to  $p = 2$ , but with no improvement in classification accuracy. With this subset model, four observations were flagged as possible outliers. Fitting the traditional GMM, we would choose a model with four groups, classifying 77% of the patients correctly. The updated Gaussian mixture model fared better at picking the correct structure, with *ICOMP* only choosing either of the correct models. The lowest *ICOMP* score was for a model with a classification rate of 71%.

### 8.4.3 Real data - Aorta

We also fit the kernel mixture model to the aorta dataset already analyzed by the GMM. Using *ICOMP*<sub>PEU</sub>, we obtained the results shown in Table 8.51. Recall that with the GMM, we used *ICOMP*<sub>PEU\_MISP</sub>, due to the clear misspecification and high dimensionality. Since it is impossible to misspecify the kernel density, we just used *ICOMP*<sub>PEU</sub>. We estimated the bandwidth matrices with  $H_k = \left(\frac{4}{p+2}\right)^{\frac{1}{n(p+4)}} \text{diag}(\hat{\Sigma}_k)$ , and used GARM to initialize GEM. As can be seen in Table 8.51, the minimum *ICOMP* value occurred when the algorithm attempted to fit  $\hat{K} = 3$  mixtures, and it obtained a better fit by dropping down to  $\hat{K} = 2$ . We were, of course, very pleased with the 100.00% rate of correct classification with this method. Even then, considering the next two

Table 8.51: Aorta data - Results from Fitting Mixtures of Kernels.

$\hat{K}$ attempted	$\hat{K}$ fit	$ICOMP_{PEU}$	Correct Classification Rate
1	1	59235.65	53.59
2	2	52674.87	55.98
<b>3</b>	<b>2*</b>	<b>49061.42</b>	<b>100.00</b>
4	2	49746.08	97.13
5	3	49532.27	99.76
6	3	49525.54	99.28

lowest scores, the classification rates were 99.28% (3 misclassified) and 99.76% (1 misclassified), respectively. Even though these models were obtained by fitting an extra population, they could still be very useful for image-based diagnosis. We would end with the confusion matrix for the best  $ICOMP_{PEU}$  model, but with 0% error, there seems to be no real point.

**SUMMARY:** We first used the EM algorithm with K-Means to apply the GMM to this dataset; it chose a model with four populations classifying 77% of the observations. We then used GEM(GARM) with the Gaussian mixture model;  $ICOMP_{PEU\_MISP}$  selected  $\hat{K} = 2$  in all ten runs. The best score was associated with a model misclassifying 8% of the patients. Using this model, we then used the GA to identify several subsets in which perfect separation of the classes was evident. With the kernel mixture model, the best three models identified by  $ICOMP_{PEU}$  correctly classified 100% ( $\hat{K} = 2$ ), 99% ( $\hat{K} = 3$ ), and 99.8% ( $\hat{K} = 3$ ).

## 8.5 PEKMM

Finally, we show results from fitting the power exponential kernel mixture model to various simulated and real-world datasets.

### 8.5.1 Simulation S4 - Nonoverlapping

The first dataset to which we applied the PE kernel mixture model is the only easy simulation presented in this research. There are  $K = 3$  nonoverlapping spherical clusters. We choose this dataset for our hybrid EM algorithm (initialized by GARM), to demonstrate its performance in the environment in which we would expect the traditional EM algorithm to do relatively well. Using the S4 protocol, we ran  $M = 25$  Monte Carlo simulations with a sample size of  $n = 200$  observations. Figure 8.22 shows the best model as identified by both  $SBC$  and  $ICOMP_{PEU}$  - only a single observation was misclassified. Regarding model selection frequencies, all four criteria ( $AIC$ ,  $SBC$ ,  $ICOMP$ , and  $ICOMP_{PEU}$ ) performed very similarly.  $AIC$  and  $ICOMP$  both selected models that correctly classified approximately 89% of the observations, but overfit the group structure. In Table 8.52, we show the estimated and actual parameters for all three populations. Especially for the first two groups, it is easy to see the tradeoff between estimating variance and kurtosis - one can increase at the expense of the other. Consider, for example, the second dimension of the first group. The estimated variance is much higher than that used to generate the data which would indicate a wider distribution, but the kurtosis parameter is also higher, which squished the distribution back into a smaller range.

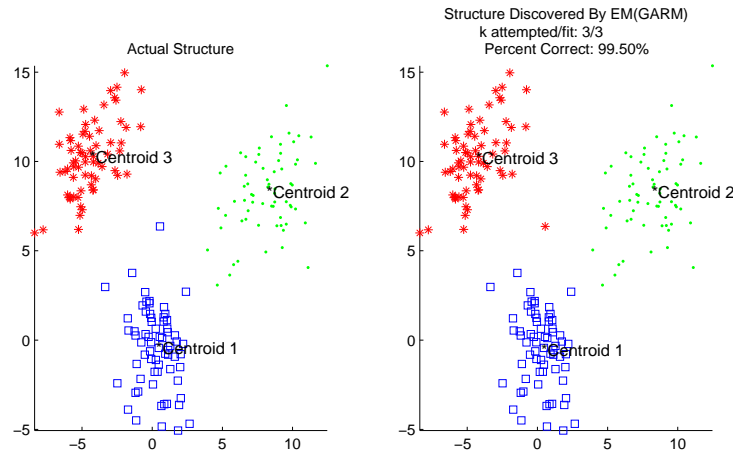


Figure 8.22: Simulation S4 - PE kernel Mixture Model Identified by  $SBC$  and  $ICOMP_{PEU}$ .

Table 8.52: Simulation S4 - Estimated and Actual Shape and Scale Parameters.

Estimated Parameters			Actual Parameters
$H_k$	$\hat{\beta}_k$	$\hat{\Sigma}_k$	$\Sigma_k$
$\begin{bmatrix} 0.76 \\ 2.10 \end{bmatrix}$	$\begin{bmatrix} 1.00 \\ 1.50 \end{bmatrix}$	$\begin{bmatrix} 1.56 & -0.58 \\ -0.58 & 4.32 \end{bmatrix}$	$\begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 0.50 \end{bmatrix}$
$\begin{bmatrix} 1.80 \\ 2.60 \end{bmatrix}$	$\begin{bmatrix} 1.50 \\ 0.83 \end{bmatrix}$	$\begin{bmatrix} 3.58 & 1.74 \\ 1.74 & 5.19 \end{bmatrix}$	$\begin{bmatrix} 4.44 & 2.17 \\ 2.17 & 6.18 \end{bmatrix}$
$\begin{bmatrix} 1.30 \\ 2.10 \end{bmatrix}$	$\begin{bmatrix} 0.85 \\ 1.20 \end{bmatrix}$	$\begin{bmatrix} 2.73 & 1.45 \\ 1.45 & 4.19 \end{bmatrix}$	$\begin{bmatrix} 3.10 & 2.00 \\ 2.00 & 4.45 \end{bmatrix}$

In the third group, we note that  $\hat{\Sigma}$  is much more similar to  $\Sigma$ . It is no surprise, then, that both of the  $\beta$ 's are closer to 1.0, since the sample was randomly drawn from a Gaussian distribution.

**SUMMARY:** Using the hybrid EM algorithm, the best model selected by *SBC* and *ICOMP<sub>PEU</sub>* only misclassified a single observation out of 200. The recovered covariance matrix parameters from this model are similar to the true values, with the expected tradeoff between estimating variances and kurtosis parameters.

### 8.5.2 Real data - Wine

The first real dataset to which we applied the PE kernel mixture model is the wine dataset of *Fiorina et al*, with  $p = 13$  variables and  $K = 3$  groups. We used GARM to initialize GEM for this dataset. From fifty replications, *AIC* picked the correct structure 42% of the time, while *SBC* did better - it picked  $\hat{K} = 3$  in 30 of the 50 simulations. Surprisingly, neither form of *ICOMP* performed well at all: *ICOMP* was minimized with a model correctly classifying 71.9% of the wines into only two groups. The heavier penalty of *ICOMP<sub>PEU</sub>*, hurt the results for this dataset - it always picked a model with no group structure. While *SBC* picked the correct structure in most replications, it was minimized (as was *AIC*) at a model with a single spurious group. The confusion matrix is shown in Table 8.53. With the exception of almost half the first cultivar, the majority of wines were placed in the correct groups; 82.58% of the observations were correctly classified.

Table 8.53: Wine data - Confusion Matrix for Best Model Selected by *SBC*.

		Predicted				
Actual	$k$	1	2	3	4	Total
	1	37	0	0	22	<b>59</b>
	2	3	65	0	3	<b>71</b>
	3	0	3	45	0	<b>48</b>
	Total	<b>40</b>	<b>68</b>	<b>45</b>	<b>25</b>	<b>178</b>

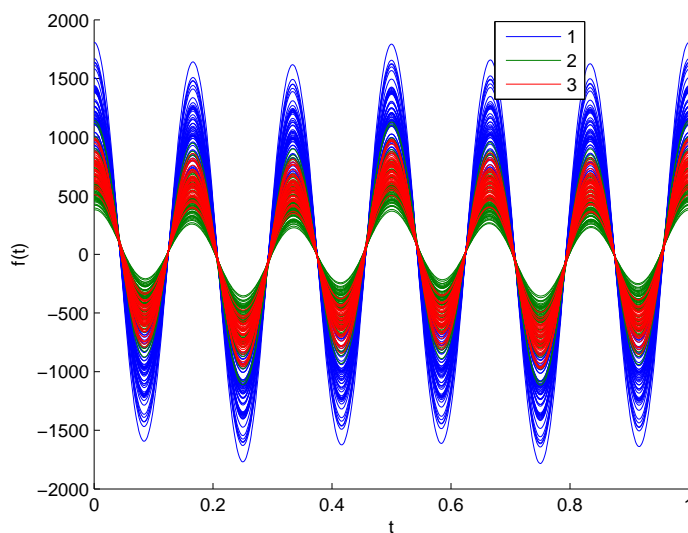


Figure 8.23: Wine data - Andrews Curves Plot.

This result suggests that some of the wines from the first cultivar were somehow quantitatively different than the others. In fact, Figure 8.23 shows the Andrews plot of the data - there seems to be a degree of separation in this group that is not exhibited by the other two cultivars. Table 8.54 shows the shape parameter  $\beta$ , independently estimated in each dimension for each group. It is clear that the different shapes in each dimension, as shown in Figures A.15 and A.16 are being modeled. Even though the model identified as best includes an extra group, we used it to perform outlier detection, using *SBC*. This procedure identified four observations as decreasing the fit of the model: one from the 3<sup>rd</sup> group, and two each from the 1<sup>st</sup> and 2<sup>nd</sup> groups. For  $p = 13$  variables, there are  $2^{13} - 1 = 8,191$  possible subsets, and we performed several runs of the GA, applying this mixture model to subsets of the original variables. The five subsets with the best *SBC* scores are shown in Table 8.55; there are several variables that show up in all five subsets. In fact, the subset with the worst top five score is included in them all. These obviously important variables are:

Table 8.54: Wine data - Estimated PE Shape Parameters for Each Dimension in Each Group.

$k$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$	$\hat{\beta}_9$	$\hat{\beta}_{10}$	$\hat{\beta}_{11}$	$\hat{\beta}_{12}$	$\hat{\beta}_{13}$
1	0.66	0.74	0.63	0.82	1.20	2.20	1.20	0.84	1.00	1.20	1.60	1.20	1.40
2	0.77	0.68	0.79	0.74	0.46	1.20	0.52	1.50	0.67	0.50	0.95	1.20	0.80
3	1.40	1.20	1.80	2.60	1.50	0.64	1.10	1.60	0.48	2.10	1.50	0.85	1.40
4	0.92	0.60	1.80	0.79	0.49	1.00	7.00	0.67	0.59	1.50	0.95	0.97	0.66

Table 8.55: Wine data - Partial Post-subset Analysis Results Summary.

Subset	$SBC$
$\{3, 6, 7, 8, 9, 11, 12\}$	-3063.62
$\{1, 3, 6, 7, 8, 9, 11, 12\}$	-3037.14
$\{3, 6, 7, 8, 11, 12\}$	-2942.55
$\{1, 3, 6, 7, 8, 11, 12\}$	-2909.27
$\{3, 6, 8, 11, 12\}$	-2787.12

$x_3$  Ash,

$x_6$  Total Phenols,

$x_8$  Non-flavonoid Phenols,

$x_{11}$  Hue,

$x_{12}$  OD280/OD315 of Diluted Wines.

We consulted with the gentleman mostly responsible for some of the best “natural” wines to come from California, about these five variables. Unfortunately, he had no specific knowledge about why they would be so important. The substantial decrease in dimensionality is especially interesting, considering the low correlations in the data (most  $< 0.5$ ). Finally, we took the top two subsets and fit the  $\hat{K} = 4$  mixture model to each. Table 8.56 shows the classification rates for the best model chosen by information criteria. Using the 2<sup>nd</sup> best subset model, all criteria selected models that correctly classified more wines than the model fit to the entire dataset. *ICOMP* chose a model that only misidentified 19 observations - 39% fewer mistakes. We show in Table 8.57 the confusion matrix for this model. Only one observation is included in the spurious group in this model, which correctly classified 89.33% of the wines. This wine is shown in the Andrews plot in Figure 8.24 as the heavier black line. It is clear that, depending upon dimension, it could fit in either group, or none. In fact, this observation was flagged as a potential outlier; it had the lowest  $SBC$  ratio.



Table 8.56: Wine data - Classification Rates from Best Subset PE Kernel Mixture models.

	$\{3, 6, 7, 8, 9, 11, 12\}$	$\{1, 3, 6, 7, 8, 9, 11, 12\}$
$AIC$	70.79%	83.15%
$SBC$	74.72%	84.83%
$ICOMP$	75.28%	89.33%
$ICOMP_{PEU}$	74.16%	85.96%

Table 8.57: Wine data - Confusion Matrix from Best Subset Mixture Model.

		Predicted					
		$k$	1	2	3	4	Total
Actual	1	59	0	0	0		<b>59</b>
	2	10	52	8	1		<b>71</b>
	3	0	0	48	0		<b>48</b>
	Total	<b>69</b>	<b>52</b>	<b>56</b>	<b>1</b>		<b>178</b>

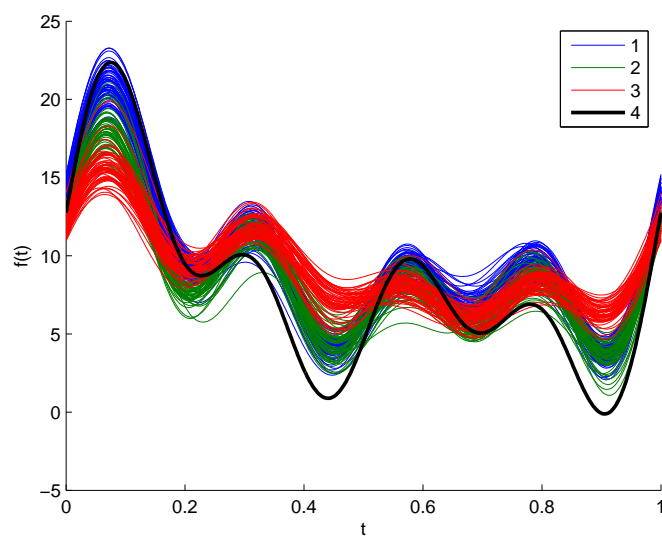


Figure 8.24: Wine data - Andrews Curves Plot from Best Subset Mixture Model.

**SUMMARY:** We ran the 50 replications of the PEK mixture model on this dataset; *SBC* selected a model with  $\hat{K} = 4$  population correctly classifying 83% of the wines. This mixture model was then used by the GA to evaluate subsets of the original variables. Despite low correlations, a subset composed of eight of the original variables, for which all criteria picked models classifying more wines than the original model. The model chosen by *ICOMP* only classified a single wine in the extra group, and put 89% of the wines in the correct groups. This observation was also flagged as an outlier, and seems to exhibit some unique cross-cultivar characteristics.

### 8.5.3 Real Data - Colon

Our final dataset is perhaps the most interesting. There is a paltry  $n = 65$  observations of  $p = 5$  measurements from  $K = 5$  groups. Figure 8.25 shows the results from Mardia's test for multivariate normality on this dataset. Both the kurtosis and skewness tests reject the null hypothesis of Gaussianity. Of course, the sample size is small, but the scatter plot matrix in the appendix, Figure A.11, seems to visually support the conclusion. Figure 8.26, created by computing the Sammon's mapping for Multi-dimensional Scaling based on the Euclidian distances of the  $(0,1)$  normalized data, shows how confounded these groups really are. Table 8.58 shows model selection frequencies and results from this dataset, using GEM(GARM). As would be expected, *AIC* exhibited the highest tendency to overfit, with *ICOMP*<sub>PEU</sub> exhibiting the opposite behavior.

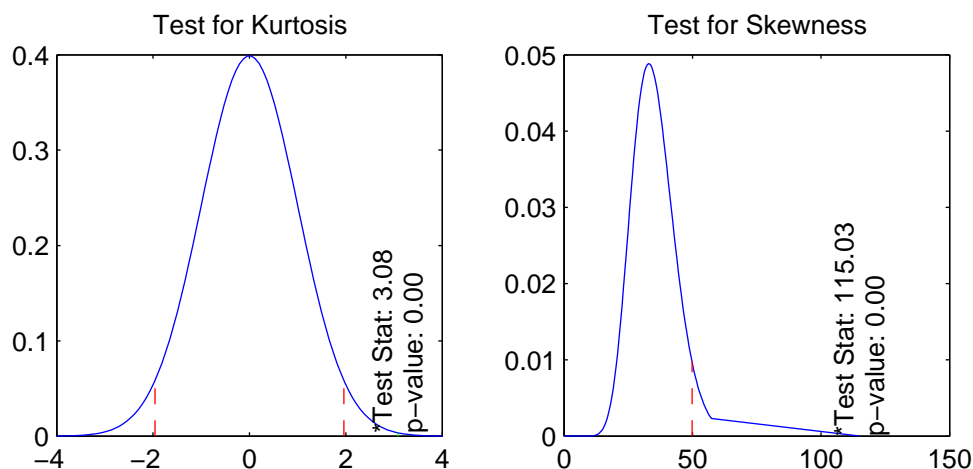


Figure 8.25: Colon data - Results from Normality Test.

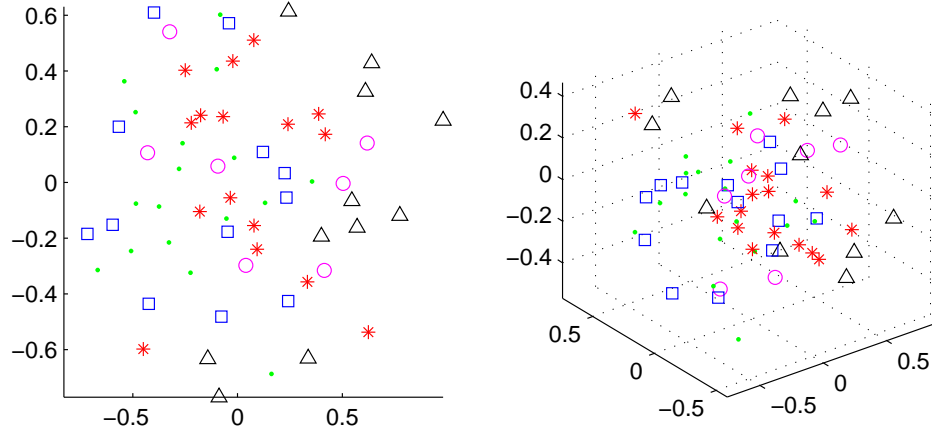


Figure 8.26: Colon data - Two- and Three- Dimensional Scatter Plots of MDS-Reduced Data.

Table 8.58: Colon data - PE Kernel Mixture Model Selection Frequencies from 10 GEM(GARM) replications.

$\hat{K}$	$AIC$	$SBC$	$ICOMP$	$ICOMP_{PEU}$
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	20.0
3	0.0	0.0	0.0	50.0
4	0.0	20.0	20.0	0.0
<b>5</b>	<b>60.0</b>	<b>60.0*</b>	<b>70.0*</b>	<b>30.0*</b>
6	40.0*	20.0	10.0	0.0
Best	38.46%	36.92%	35.39%	43.08%

While a probability of correct classification less than 50% seems very poor, and a far cry from all other results shown here, consider there are five completely confounded groups. Random classification (guessing) would not be expected to do any better than 20%. Now recall how small the sample was - perhaps this performance is not so bad after all. The smallest true population, in fact, had a mere seven observations. In Figure A.11 in the appendix, we see no dimensions showing a high peak. The estimated  $\beta$  parameters in Table 8.59 show the same characteristics. Using the best  $ICOMP_{PEU}$  model, we detected two possible influential observations in this dataset, as shown in Figure 8.27. We then fit this same mixture model to all nontrivial 31 subset models, and identified two subsets for further analysis -  $\{x_1, x_5\}$  ( $-325.62$ ) and  $\{x_1, x_2, x_5\}$  ( $-168.53$ ); the  $ICOMP_{PEU}$  scores (in parentheses) are much lower than that for the saturated model - 129.92. Using the 2<sup>nd</sup> subset model,  $SBC$  identified a mixture model that misclassified an additional single

Table 8.59: Colon data - Estimated PE Shape Parameters from Best  $ICOMP_{PEU}$  Model.

$k$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$
1	1.30	10.0	7.70	4.20	4.30
2	1.00	1.30	1.40	1.00	10.0
3	1.20	10.0	3.50	1.30	3.70
4	10.0	10.0	10.0	10.0	10.0
5	1.10	2.10	1.30	1.00	10.0

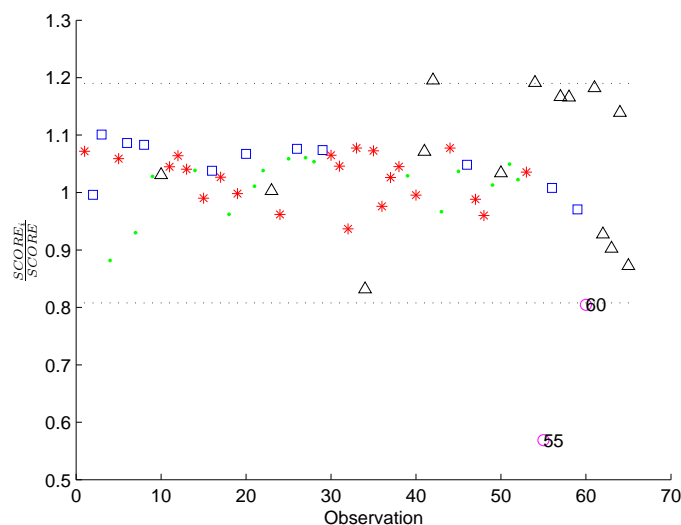


Figure 8.27: Colon data - Identifying Two Possible Influential Observations.

observation, though we reduced the dimensionality of the dataset by two variables the classification rates for these two subset models are shown in Table 8.60. In Table 8.61, we see that the highest correlation was between  $x_2$  (kept) and  $x_4$ , which was dropped. The next highest correlation was between  $x_5$ , which was kept, and  $x_3$  (dropped). The remainder of the correlations are basically negligible.

Table 8.60: Colon data - Classification Rates from Best Subset PE Kernel Mixture models.

	$\{1, 5\}$	$\{1, 2, 5\}$
$AIC$	36.92%	29.23%
$SBC$	35.38%	41.54%
$ICOMP$	29.23%	29.23%
$ICOMP_{PEU}$	32.31%	32.31%

Table 8.61: Colon data - Correlation Matrix.

	$x_1$	$x_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$x_5$
$x_1$	1.00	-0.162	<b>-0.338</b>	<b>-0.120</b>	-0.361
$x_2$		1.00	<b>0.303</b>	<b>0.799</b>	0.076
$\mathbf{x}_3$			1.00	<b>0.148</b>	<b>0.538</b>
$\mathbf{x}_4$				1.00	<b>0.060</b>
$x_5$					1.00

**SUMMARY:** For this very small and complex dataset,  $ICOMP$  selected a model with the true grouping structure in seven of the ten replications of fitting the PEK mixture model. The best  $ICOMP_{PEU}$  model correctly classified 43% of the observations - twice as good as random classification (guessing). Given how confounded the groups are, this is very good performance. We then evaluated the 31 subset models, and identified one with three variables, for which  $SBC$  identified a  $\hat{K} = 5$  mixture model which only classified a single additional observation.

# Conclusions

*“It’s a mat, with **conclusions**, that you can ... jump to.”* - Tom Smykowski, Office Space

## 9.1 Summary of Dissertation

In this dissertation, we have developed and merged several newer and cutting-edge statistical techniques to modernize mixture modeling and expand the power and usefulness of this method of unsupervised classification. Problems we simultaneously address include:

- Misspecified functional form,
- High dependence upon initial values,
- Numerical instability of the covariance matrix,
- Intelligent selection of a most appropriate model, and
- Simultaneous outlier detection and dimension reduction.

Chapters 1 and 2 introduced mixture modeling and reviewed its history all the way back to the 19<sup>th</sup> century. Chapter 3 began with some background and derivation of information criteria, first introduced to the mixture problem by Bozdogan (1983). We finished Chapter 3 with the problem of covariance singularity, and our proposed solution - the use of newer robust covariance estimators that augment the MLEs for each population independently. The last of the “background” chapters, Chapter 4, was dedicated to the genetic algorithm, and detailed several specialized variants. Alternative optimization algorithms were also discussed.

Chapters 5, 6, and 7 constitute the bulk of the new research. In 5, we discuss symmetric elliptically-contoured distributions and some of the literature results. We then provided details of a hybrid EM algorithm for the EC mixture model, and how to compute various information criteria under this distributional specification. Chapter 6 similarly reviews existing research and presents our new details regarding fitting a mixture of multivariate kernel density estimators. Finally we introduced our power exponential product kernel in Chapter 7, as a way to fit peak and tail behavior in each dimension and for each group independently. Chapter 8 showed results from the EC, kernel, and PEK mixture models on a variety of challenging simulated and real datasets. With many of the datasets, we compared our results to the traditional Gaussian mixture model and / or the updated GMM (with GARM, GEM, information criteria, robust covariance estimators). For all datasets, our methods exhibited either similar or superior performance to the more restrictive Gaussian mixture model.

The traditional methods are undeniably faster than the new methods. However, what good is the ability to evaluate, if the methods used are unable to consistently produce results for all models tested? What about fitting mixtures to real data in which we don't actually have any *a priori* information about the group structure, against which to judge the importance of non-convergence?

## 9.2 Future work

There are several directions in which future research along these lines could be performed. For researchers in computer science and statistical computing, we use methods that could undeniably be made more efficient. For example, relatively little is known about the genetic algorithm - its convergence rates are not analytically estimable, and little is known of its general robustness against specific parameter values. Empirical studies on these characteristics, and others, of this stochastic search algorithm would probably be of value. Additionally, other stochastic or automata-based algorithms could be considered in the context of mixture modeling. Examples include Artificial Neural Networks (ANN), the Artificial Bee Colony (ABC) optimization algorithm, or the Touring Ant Colony Optimization (TACO) algorithm.

A second avenue of further research that would be profitable would be that of even further generalizing the EC mixture model. With the ECMM, we use a functional form to adapt to the peak

and tail behavior of datasets, but still require symmetry. There are many real datasets that exhibit asymmetry - financial data as an example. Much research has been done in the area of skewing distributions, with several papers included in Ed. M Genton (2004). The ability to simultaneously fit peak, tail, and skew behavior with a distribution could hold much promise. Additionally, recall how we simplified the IFIM for the EC distribution. Deriving the fully-specified model covariance matrix including the pdf generator-specific shape parameters would undoubtedly be invaluable. In a similar vein, we could develop a skew PE product kernel mixture model, in which the peak, tail, and skew behavior can be tailored to each dimension of a dataset independently, using the modified bandwidth as mentioned, without imposing any functional form.

Finally, we would go back to a comment made at the beginning of Chapter 9. We commented on how rather unnatural it was, from a certain perspective, to measure models in relation to class labels that were defined based on probably partial information. For example, with the Aorta data, patients were classified as either sick or healthy - no transition group as in the diabetic data. Perhaps there truly were four groups to the aorta data - sick male, sick female, healthy male, healthy female - all with significant quantitative differences. However, since the original researcher chose not to classify the data this way, we only have two groups. Recall that, when we fit the PEK mixture model to the entire wine dataset, the wine from the first cultivar was split into two groups. Perhaps the vintners were unaware that this first group was composed of two genetically distinct grapes? Thus, we would suggest the need for **easily-interpretable**, **justifiable**, and **objective** criteria for comparing different clustering models independent of stated class labels which may understate the true underlying heterogeneity.

### 9.3 Expected Publications

Howe, A. and Bozdogan, H. (2009). Simultaneous Model Selection in Multivariate Mixture-Model Cluster Analysis Using Information Complexity and Genetic Algorithm:  $M^3$ . In Bozdogan, H., editor, *HDM 2008 Conference Book*. Chapman & Hall / CRC. (not yet published).

Howe, A. and Bozdogan, H. (2009). Multivariate Mixture Modeling on the Edge - Way Beyond Normalcy. *tbd.* (not yet published).



# Bibliography

- Aeberhard, S., Coomans, D., and de vel, O. (1992). Comparison of Classifiers in High Dimensional Settings. Technical Report 92-02, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland. 174
- Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. In Petrox, B. and Csaki, F., editors, *Second International Symposium on Information Theory.*, pages 267–281, Budapest. Academiai Kiado. vi, 16
- Anderson, D. and Steen, C. (1994). *Broadcasting and Cable Year Book.* 46
- Anderson, T. and Fang, K. (1990). Inference in Multivariate Elliptically Contoured Distributions Based on Maximum Likelihood. In Fang, K. and Anderson, T., editors, *Statistical Inference in Elliptically Contoured and Related Distributions.*, pages 201–216. Allerton Press, Inc., New York. v, 50, 51
- Andrews, D. and Herzberg, A. (1985). *DATA.* Springer-Verlag, New York. 101, 171
- Bandyopadhyay, S. (2005). Simulated Annealing Using a Reversible Jump Markov Chain Monte Carlo Alorithm for Fuzzy Clustering. *IEEE Transations on Knowledge and Data Engineering*, 17(4):479–490. 3
- Bensmail, H. and Bozdogan, H. (2002). Regularized Kernel Discriminant Analysis with Optimally Scaled Data. In Nishisato, S., Baba, Y., Bozdogan, H., and Kanefuji, K., editors, *Measurement and Multivariate Analysis*, pages 133–144. Springer, Tokyo, Japan. 74
- Bernardo, J. and Girón, J. (1988). A Bayesian Analysis of Simple Mixture Problems. In J.M. Bernardo, M.H. DeGroot, D. L. and Smith, A., editors, *Bayesian Statistics 3*, pages 67–78. Oxford University Press, Oxford. 3
- Bhandarkar, S., Zhang, Y., and Potter, W. (1994). n Edge Detection Technique Using Genetic Algorithm-Based Optimization. *Pattern Recognition*, 27:1159–1180. 48
- Bhuyan, J., Raghavan, V., and Elayavalli, V. (1991). Genetic Algorithm for Clustering with an Ordered Representation. In *4th International Conference on Genetic Algorithms*, San Mateo, CA. Morgan Kaufman. vi, 38
- Box, G. (1979). *Robustness in Statistics.* Academic Press, London. 21

- Bozdogan, H. (1981). *Multi-Sample Cluster Analysis and Approaches to Validity Studies in Clustering Individuals*. PhD thesis, University of Illinois at Chicago. 3, 24
- Bozdogan, H. (1983). Determining the Number of Component Clusters in the Standard Multivariate Normal Mixture Model Using Model-Selection Criteria. Technical Report UIC/DQM/A83-1, University of Illinois at Chicago, Quantitative Methods Department, Illinois. ARO Contract DAAG29-82-K-0155. 2, 9, 138
- Bozdogan, H. (1988). ICOMP: A New Model-Selection Criteria. In Bock, H., editor, *Classification and Related Methods of Data Analysis*. North-Holland. 16, 18
- Bozdogan, H. (1990). On the Information-Based Measure of Covariance Complexity and its Application to the Evaluation of Multivariate Linear Models. *Communication in Statistics, Theory and Methods*, 19:221–278. 19
- Bozdogan, H. (1994a). Choosing the Number of Clusters, Subset Selection of Variables, and Outlier Detection in the Standard Mixture-Model Cluster Analysis. In Diday, E., editor, *New Approaches in Classification and Data Analysis*, pages 169–177. Springer-Verlag, New York. vi, 3, 44
- Bozdogan, H. (1994b). Mixture-Model Cluster Analysis Using Model Selection Criteria and a New Informational Measure of Complexity. In Bozdogan, H., editor, *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach*, volume 2, pages 69–113, Dordrecht, the Netherlands. Kluwer Academic Publishers. vi, 3, 8, 24
- Bozdogan, H. (1995). Statistical Modeling and Model Evaluation: a New Informational Approach. A forthcoming book. 53, 54
- Bozdogan, H. (2000). Akaike’s Information Criterion and Recent Developments in Information Complexity. *Journal of Mathematical Psychology*, 44:62–91. 22, 25
- Bozdogan, H. and Haughton, D. (1998). Informational Complexity Criteria for Regression Models. *Computational Statistics and Data Analysis*, 28:51–76. 19
- Burns, P. (1992). A Genetic Algorithm for Robust Regression Estimation. Statsci technical report from Statistical Sciences, Inc. 48
- Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society, Series A*, 158:419–466. 20

- Chen, J., Ching, R., and Lin, Y. (2004). An Extended Study of the K-Means Algorithm for Data Clustering and Its Applications. *The Journal of the Operational Research Society*, 9:976–987. 3
- Chen, M. (1976). Estimation of Covariance Matrices Under a Quadratic Loss Function. Research Report S-46, Department of Mathematics, SUNY at Albany. 29
- Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press, Princeton, New Jersey. 22
- Day, N. (1969). Estimating the Components of a Mixture of Normal Distributions. *Biometrika*, 56:463–474. 3
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38. 11
- Diebolt, J. and Robert, C. (1994). Estimation of Finite Mixture Distributions through Bayesian Sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(2):363–375. 3
- Ed. M Genton (2004). *Skew-Elliptical Distributions and Their Applications*. Chapman & Hall / CRC, Boca Raton, Florida. 67, 140
- Efron, B. and Hinkley, D. (1978). Assessing the Accuracy of the Maximum Likelihood Estimator: Observed Versus Expected Fisher Information. *Biometrika*, 65(3):457–482. 19
- Fang, K., Kotz, S., and Ng, K. (1990). *Symmetric Multivariate and Related Distributions*. Chapman and Hall, New York. v, 50
- Farrell, M. and Mersereau, R. (2004). Estimation of Elliptically Contoured Mixture Models for Hyperspectral Imaging Data. In *Geoscience and Remote Sensing Symposium, IGARSS '04*, volume 4, pages 2412–2415. IEEE International. v, 52, 53, 63
- Feng, Z. and McCulloch, C. (1994). On the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture with Unequal Variances. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:609–617. 3
- Fernandez, C. and Steel, M. (1998). On Bayesian Modeling of Fat Tails and Skewness. *Journal of the American Statistical Association*, 93(441):359–371. 70

- Fiebig, D. (1982). *The Maximum Entropy Distribution and Its Covariance Matrix*. PhD thesis, University of Southern California. 29
- Hamada, M., Martz, H., Reese, C., and Wilson, A. (2001). Finding Near-Optimal Bayesian Experimental Designs via Genetic Algorithms. *The American Statistician*, 55(3):175–181. 48
- Hartigan, J. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York. 3
- Hasselblad, V. (1966). Estimation of Parameters for a Mixture of Normal Distributions. *Technometrics*, 8(3):1459–1471. v, 3
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan. 2, 31
- Holland, J. (1992). Genetic Algorithms. *Scientific American*, pages 66–72. 31
- Jain, A. and Dubes, R. (1989). *Algorithms for Clustering Data*. Prentice Hall, New Jersey. 11
- John, S. (1970a). On Identifying the Population of Origin of Each Observation in a Mixture of Observations from Two Gamma Populations. *Technometrics*, 12(3):565–568. v, 3
- John, S. (1970b). On Identifying the Population of Origin of Each Observation in a Mixture of Observations from Two Normal Populations. *Technometrics*, 12:553–563. 3
- Kabir, A. (1968). Estimation of Parameters of a Finite Mixture of Distributions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(3):472–482. 3
- Klein, R. and Dubes, R. (1989). Experiments in Projection and Clustering by Simulated Annealing. *Pattern Recognition*, 22(2):213–220. 3, 45
- Krishna, K. and Murty, M. (1999). Genetic K-Means Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 29(3):433–439. vi, 2, 11, 38, 41
- Kullback, A. and Leibler, R. (1951). On Information and Sufficiency. *Annals of Mathematical Statistics*, 22:79–86. 17
- Ledoit, O. and Wolf, M. (2003). Honey, I Shrunk the Sample Covariance Matrix. Technical report, Universitat Pompeu Fabra. vi, 29
- Liang, F. and Wong, W. (2001). Real-Parameter Evolutionary Monte Carlo with Applications to Bayesian Mixture Models. *Journal of the American Statistical Association*, 96(454):653–666. 48

- Lindley, D. (1956). On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005. 19
- Liu, M. (2006). *Multivariate Nonnormal Regression Models, Information Complexity, and Genetic Algorithms: A Three Way Hybrid for Intelligent Data Mining*. PhD thesis, The University of Tennessee, Knoxville. 54, 55, 64
- Liu, S. (2002). Local Influence in Multivariate Elliptical Linear Regression Models. *Linear Algebra and its Applications*, 354:159–174. 54
- Ma, J. and Xu, L. (2005). Asymptotic Convergence Properties of the EM Algorithm with Respect to the Overlap in the Mixture. *Neurocomputing*, 68:105–129. 3, 13
- MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA. University of California, Berkeley. 2, 9
- Mallows, C. (1973). Some comments on cp. *Technometrics*, 15:661–675. 23
- Mao, J. and Jain, A. (1996). A Self-Organizing Network for Hyperellipsoidal Clustering (HEC). *IEEE Transactions on Neural Networks*, 7:17–29. 3, 40, 41
- Mardia, K. (1974). Applications of Some Measures of Multivariate Skewness and Kurtosis in Testing Normality and Robustness Studies. *Sankhya*, B36:115–128. 102
- Marriott, F. (1971). Practical Problems in a Method of Cluster Analysis. *Biometrics*, 27(3):501–514. 3
- McLachlan, G. (1987). On Bootstrapping the Likelihood Ratio Test Statistic for the Number of Components in a Normal Mixture. *Applied Statistics*, 36:318–324. 3
- Meyer, M. (2003). An Evolutionary Algorithm with Applications to Statistics. *Journal of Computational and Graphical Statistics*, 12(2):265–281. 48
- Neely, C., Weller, P., and Dittmar, R. (1997). Is Technical Analysis in the Foreign Exchange Market Profitable? A Genetic Programming Approach. *Journal of Financial and Quantitative Analysis*, 32(4):405–426. 48

- Pearlman, J. (1986). *Nuclear Magnetic Resonance Spectral Signatures of Liquid Crystals in Human Atheroma As Basis For Multi-Dimensional Digital Imaging of Atherosclerosis*. PhD thesis, University of Virginia. 107, 166
- Pearson, K. (1894). Contributions to the Mathematical Theory of Evolution. In *Phil. Trans. Royal Society*, volume 185A, pages 71–110. v, 2, 6
- Pérez, J. and Berger, J. (2002). Expected-Posterior Prior Distributions for Model Selection. *Biometrika*, 89(3):491–511. 3
- Peters, B. and Walker, H. (1978). An Iterative Procedure for Obtaining Maximum-Likelihood Estimates of the Parameters for a Mixture of Normal Distributions. *SIAM Journal on Applied Mathematics*, 35(2):362–378. 3, 9
- Poskitt, D. (1987). Precision, Complexity and Bayesian Model Determination. *Journal of the Royal Statistical Society, Series B (Methodological)*, 49(2):199–208. 19
- Press, S. (1975). Estimation of a Normal Covariance Matrix. Technical report, University of British Columbia. 29
- Rao, C. (1945). Information and Accuracy Attainable in the Estimation of Statistical Parameters. In *Bulletin of the Calcutta Math Society*, volume 37, page 81. 22
- Rao, C. (1947). Minimum Variance and the Estimation of Several Parameters. In *Proceedings of the Cambridge Philosophical Society*, volume 43, page 280. 22
- Rao, C. (1948). Sufficient Statistics and Minimum Variance Estimates. In *Proceedings of the Cambridge Philosophical Society*, volume 45, page 213. 22
- Rechardson, S. and Green, P. (1997). On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Journal of the Royal Statistical Society. Series B (Methodological)*, 59:731–792. 4
- Reddy, C. and Chiang, H. (2007). Component-wise density smoothing for parameter estimation of mixture models. 76
- Redner, R. and Walker, H. (1984). Mixture Densities, Maximum Likelihood and the EM Algorithm. *SIAM Review*, 26(2):195–239. 3, 11, 12

- Rider, P. (1961). The Method of Moments Applied to a Mixture of Two Exponential Distributions. *The Annals of Mathematical Statistics*, 32(1):143–147. 3
- Rosenblatt, M. (1956). Remarks on Some Nonparametric Estimates of a Density Function. *The Annals of Mathematical Statistics*, 27:832–837. 67
- Routledge, B. (1999). Adaptive Learning in Financial Markets. In *The Review of Financial Studies*, volume 12, pages 1165–1202. Oxford University Press. 48
- Schwarz, G. (1978). Estimating the Dimension of a Model. *Annals of Statistics*, 6:461–464. vi, 18
- Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, New York. 68, 70, 73
- Seo, T. and Toyama, T. (1996). On The Estimation of Kurtosis Parameter in Elliptical Distributions. *Journal of Japanese Statistical Society*, 26:59–68. 53
- Shurygin, A. (1983). The Linear Combination of the Simplest Discriminator and Fisher’s One. In Nauka, editor, *Applied Statistics*. Moscow, Russia. vi, 29
- Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London. 70, 73
- Song, W., Feng, M., Wei, S., and Shaowei, X. (1997). The Hyperellipsoidal Clustering Using Genetic Algorithm. In *1997 IEEE International Conference on Intelligent Processing Systems*, pages 592–596. 3, 41, 42
- Song, W. and Shaowei, X. (1997). Comments on “A Self-organizing Network for Hyperellipsoidal Clustering (HEC)”. *IEEE Transactions on Neural Networks*, 8:1561–1562. 3, 41
- Stephens, M. (2000). Bayesian Analysis of Mixture Models with an Unknown Number of Components - an Alternative to Reversible Jump. *The Annals of Statistics*, 28(1):40–74. 4
- Street, W., Wolberg, W., and Mangasarian, O. (1993). Nuclear Feature Extraction for Breast Tumor Diagnosis. volume 1905, pages 861–870. 90, 168
- Sutradhar, B. and Ali, M. (1986). Estimation of the Parameters of a Regression Model with a Multivariate T Error Variable. *Communication in Statistics, Theory and Methods*, 15(2):429–450. 52, 63



- Takeuchi, K. (1976). Distribution of Information Statistics and Criterion of Model Fitting. *Suri-Kagaku (Mathematical Sciences)*, 153:12–18. In Japanese. 22
- Theil, H. and Fiebig, D. (1984). *Exploiting Continuity: Maximum Entropy Estimation of Continuous Distributions*. Ballinger Publishing Company, Cambridge, Massachusetts, USA. 29
- Thomaz, C. (2004). *Maximum Entropy Covariance Estimate for Statistical Pattern Recognition*. PhD thesis, University of London and for the Diploma of the Imperial College (D.I.C.). vi, 29
- Van Emden, M. (1971). An Analysis of Complexity. In *Mathematical Centre Tracts.*, volume 35. Mathematisch Centrum. 15
- Vose, M. (1999). *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press. 31
- Wand, M. and Jones, M. (1993). Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation. *Journal of the American Statistical Association*, 88(422):520–528. 73
- Wand, M. and Jones, M. (1995). *Kernel Smoothing*. Chapman & Hall, London. v, 73
- Wang, J., Lee, J., and Zhang, C. (2003a). Kernel GMM And Its Application to Image Binarization. In *Proc. of ICME2003*. 4
- Wang, J., Lee, J., and Zhang, C. (2003b). Kernel Trick Embedded Gaussian Mixture Model. *Algorithmic Learning Theory*, 2842:159–174. 4
- Wegman, E. (1986). Hyperdimensional Data Analysis Using Parallel Coordinates. Technical Report #1, George Mason University Center for Computational Statistics. 171
- West, J. and Linster (2003). The Evolution of Fuzzy Rules in Two-Player Games. *Southern Economic Journal*, 69(3):705–717. 48
- Wicker, J. (2006). *Applications of Modern Statistical Methods to Analysis of Data in Physical Science*. PhD thesis, The University of Tennessee, Knoxville. vi, 3, 42, 43
- Wolfe, J. (1971). A Monte Carlo Study of Sampling Distribution on the Likelihood Ratio Test for Mixtures of Multinormal Distributions. Technical Bulletin STB 72-2, U.S. Naval Personnel and Training Research Laboratory, San Diego, CA. 3
- Wolpert, D. and Macready, W. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82. 48

- Woo, M. and Sriram, T. (2006). Robust Estimation of Mixture Complexity. *Journal of the American Statistical Association*, 101:1475–1486. 4
- Xu, L. and Jordan, M. (1996). On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8:129–151. 3, 12
- Zhang, X., King, M., and Hyndman, R. (2004). Bandwidth Selection for Multivariate Kernel Density Estimation Using MCMC. Working paper, Monash University, Department of Econometrics and Business Statistics. 74

# Appendices

## Appendix 1: MATLAB Toolbox - M<sup>3</sup>

The M<sup>3</sup> Matlab toolbox is a flexible platform for performing statistical mixture modeling and model selection with information criteria. The toolbox, currently at version 4.20, is fully equipped to handle everything demonstrated in this dissertation. Additionally, it was designed with the flexibility to be easily extended to handle other types of mixture problems. The remainder of this short introduction to M<sup>3</sup> will be divided into three sections: Data Input, M<sup>3</sup> User Interface, and Result Output.

### Data Input

M<sup>3</sup> can model three types of data – simulated data sampled from a mixture of known distributions, real data with a known class structure, and data with no known structure. All simulated data samples are generated from the multivariate power exponential distribution (7.1). To model simulated data, point M<sup>3</sup> to a tab-delimited file matching the format shown in Figure A.1.

```
1 3 number of clusters
2 0.7 0.33^ * kurtosis parameter
3 0 0^ ^ mixing proportion
4 2 -0.65 # + mean vector
5 -0.65 1 # covariance matrix
6 1.25* 0.33^
7 4 2^
8 2 0.84 #
9 0.84 1
10 2* 0.33^
11 2 2^
12 1 -0.5 #
13 -0.5 2
```

Figure A.1: Simulated Data Format.

In this format, the first row identifies the number of mixtures which should be simulated. After that, four entries per group are required. For cluster  $k$ , the first row should identify the kurtosis parameter  $\beta_k$ , then the mixing proportion  $\pi_k$  (separated by a tab). The second row should contain the  $p$  – dimensional mean vector  $\mu_k$ , while the next  $p$  rows are the variance-covariance matrix  $\Sigma_k$ . If  $p = 2$ , M<sup>3</sup> creates and saves nice bivariate scatter plots comparing the actual and estimated structures, as already seen.

If the researcher desires to model data with a known class structure,  $M^3$  will expect a tab-delimited file with mixture identifiers ( $k = 1 \dots K$ ) in the first column, and the actual measurements (one observation per row) in the remaining columns. Note that, in the case of data with two groups, observations must be identified as belonging to cluster 1 or 2, not 0 or 1. For real data with no identified structure, all is required is a (yet again) tab-delimited file with basically the same structure as the known data. Of course, the first column will be data, and not identifiers. In all cases, the data file must be completely numeric and human-readable; nonnumeric and/or binary data will not be handled. The decision to use tab-delimited files was driven by ease of use and portability.

## $M^3$ User Interface

As can be seen in Figure A.2, the user interface for the Multivariate Mixture Modeler is organized into several sections. Each will be discussed in what follows; `%M3%` indicates the root directory in which the  $M^3$  files are installed.

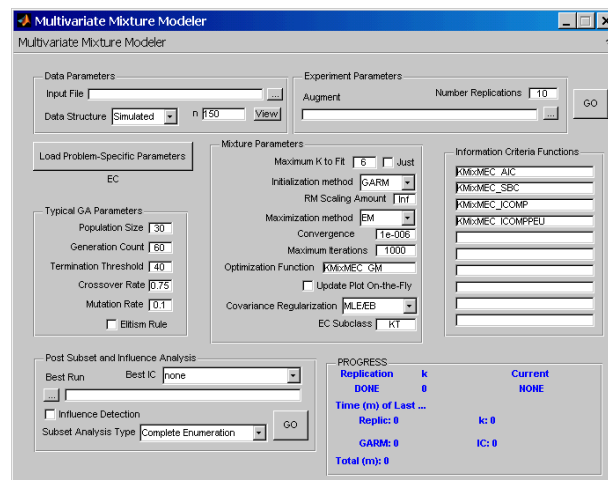


Figure A.2:  $M^3$  Toolbox GUI.

- DATA PARAMETERS

**Input File:** This is the filename for the data to be modeled. The file must be located in `%M3%\data\`. The simplest way to fill this is to click the “...” button, which will open the familiar Windows open file dialog box.

**Data Structure:** Select from the dropdown box what type of data will be used for modeling: Simulated, Known, or Unknown.

**Number Observations:** If the Data Structure is Simulated, this parameter will be displayed.

Here the researcher can define the total number of observations in the dataset. Each mixture simulated will contain  $n_k = (\text{Number Observations}) \times \pi_k$  samples, possibly rounded.

View: If the %M3%\data\folder holds an image file related the current Input File named identically, but with an .eps extension, clicking this button will cause Windows to open the file using whatever is the default program for .eps files.

- EXPERIMENT PARAMETERS

Number Replications: The number input here determines how many times the entire modeling process is performed. At the end, M<sup>3</sup> will create and save a summary table of all the replications. If the Data Structure is Simulated, the data will be regenerated from the provided parameters with each replication, so Monte-Carlo simulations can be performed.

Augment: After each replication, M<sup>3</sup> saves a Matlab workspace (binary) file named according to the mask “MIXTMR\_”%Input File%%Timestamp%.mat. This file stores all the parameters used, as well as results from the replications. When it is finished, the filename is placed in this box for convenience. The Augment feature is very convenient. Consider a case in which the researcher executes 10 Monte-Carlo simulations as an exploratory step, then decides he wants to perform a larger experiment. He doesn’t need to start over from 0. All that is required is to click the “...” button and locate the specific experiment .mat file from the run to augment, then type 100 (or however many he wants) in the Number Replications box and hit “GO”. M<sup>3</sup> will perform the modeling ninety more times, combine them with the previous ten, and summarize all 100 as if they were run sequentially. Note that, no matter what parameters the researcher may have set in the interim, loading a experiment to augment will set everything to the same parameters previously used. Changing parameters after loading an experiment for augmentation will likewise have no effect. If an experiment is loaded to augment, and the user changes his mind, there’s no need to quit the toolbox just to clear the memory - select Reset in the Multivariate Mixture Model menu, or hit CTRL+R.

- TYPICAL GA PARAMETERS

Population Size: This determines the number of chromosomes per generation for GARM, GKM, and GEM.

Generation Count: Except in cases of early termination, all genetic algorithm procedures will iterate through this many generations.

Termination Threshold: This identifies the minimum number of generations each GA procedure must perform. If the GA iterates through this many generations with no improvement in the objective function, it is deemed to have converged on a solution, and terminates prematurely.

Crossover Rate: This is the probability that determines which percentage of mating pairs actually produce crossed offspring, as opposed to genetic replication.

Mutation Rate: This percentage is used to determine both the rate at which chromosomes mutate after mating and the probability of individual loci mutating. This same rate applies to all GA procedures.

Elitism Rule: If the Elitism Rule is on, at the end of each GA generation, the most fit chromosome is copied directly into the new generation with no modification.

- MIXTURE PARAMETERS

Maximum K to Fit: The default behavior of  $M^3$  is to attempt to fit  $1 \dots \hat{K}$  mixtures to the data; If the *Just* box is checked, it will attempt to fit exactly  $\hat{K}$  mixtures. Note that, while GARM and GKM are required to prevent any clusters from dropping out, the EM and GEM algorithms are not. Thus, the researcher could try to fit  $\hat{K} = 5$  mixtures to some dataset, and have the final result really be  $\hat{K} = 3$  groups, as already seen.

Initialization Method: Choose between the traditional K-Means algorithm, GARM, or GKM.

RM Scaling Amount: The regularized Mahalanobis distance is computed as in (4.13); the value of the exponent  $c$  is taken from this box. A logical scaling value is  $c = 0.5$ , since  $|\hat{\Sigma}_k|^{\frac{1}{2}}$  is the square root of the generalized variance. The researcher can also scale  $m_i(k)$  by the complexity of the estimated covariance matrix  $C_1(\Sigma_k)$ . To do so, set the RM Scaling Amount to “Inf”.

Optimization Method: The  $M^3$  toolbox allows the researcher to choose between two optimization procedures - EM or GEM.

Convergence Criteria: If the Optimization Method is set to EM, this option is available. It determines the maximum difference in the log-likelihood between successive iterations allowed for convergence. This, along with Maximum Iterations, also applies to the K-Means initialization.

Maximum Iterations: Also only for EM algorithm, this identifies how many expectation-maximization iterations through which the algorithm is allowed to proceed. If Maximum Iterations are performed with no convergence, the algorithm terminates, signaling non-convergence.

Optimization Function: This is where the researcher can identify the function used to implement the Optimization Method. If Optimization Method is changed, don't forget to change Optimization Function, and visa versa.

Update Plot On-the-Fly: The functions that implement the EM and GEM algorithms each produce a progress plot. They should show convergence of their respective objective functions. It is typical for GA implementations to show a progress plot of the average and optimum objective values per generation. If this box is checked, these plots will be updated and displayed iteration-by-iteration. Otherwise, the plots are not created until the end of the procedure. Having this on will slow down  $M^3$ . Note that at the end of the EM and GEM functions, the progress plot is redrawn, saved, then closed immediately. With this box unchecked, the researcher will not be able to view the progress plots except by opening the saved file.

Covariance Regularization:  $M^3$  needs to know which smoothed covariance estimator to use; this is where the researcher can make a selection.

EC Subclass: If the elliptically-contoured mixture model is to be used, this box will appear. Choices are "KT" or "PVII" for Kotz or Pearson type VII, respectively. If however, the kernel problem-specific parameters are loaded, this box will be labeled as shown next.

Bandwidth Estimator Type: What is required is the numeric code which defines how the bandwidth matrices  $H_k$  are estimated. See Table 6.2 for details about the numeric codes.

- INFORMATION CRITERIA FUNCTIONS

- After executing the Initialization Method for a specific  $k$ , the Optimization Function will be run, and results tabulated, using each information criterion (IC) function listed here (unless it's set to EM). All functions must be available in the Matlab path, or in the %M3% directory. Of course, the function can return any number of outputs, but the first is assumed to be the IC score and is the only one taken.

- PROBLEM-SPECIFIC PARAMETERS



M<sup>3</sup> is a flexible mixture modeling platform that can readily be extended to perform under different distributional assumptions. Each of the four types of mixture modeling demonstrated in this dissertation have a file listing parameter values specific to the method. Logically, the Gaussian parameters are loaded by default. To load something else, click on this button, and navigate to a pre-defined .mat file that has the parameter values specific to the type of mixture modeling performed. M<sup>3</sup> will expect at least three variables in this file:

**PStype:** string with the problem-specific type name

**InfCrit:** (10 × 1) cell array with entries specifying the name(s) of up to ten information criteria functions

**em\_func:** string specifying the name of the optimization function

As well as these three parameters, there could be others depending upon what is specific to that problem. For example, the KernelParams.mat file also stores the variable **htype** which fills in Bandwidth Estimator Type. The string in **PStype** from the currently loaded problem-specific parameter file is displayed immediately below the selection button.

- POST SUBSET AND INFLUENCE ANALYSIS

After running M<sup>3</sup> on some data, the researcher can come to this section to identify the best subset of the data or to perform influential observation detection. Clicking the “GO” button will perform the requested analysis after selections have been made.

**Best Run:** Click on the “...” button to get the familiar Windows open file dialog box to find the .mat file from the replication you want to analyze. For each IC function, the experiment summary lists the name of the output file for the replication with the best score. An example would be GKM+GEM\_20070225\_204142.mat - this file would hold the information used for this analysis.

**Best IC:** This dropdown box is populated with the specific IC functions used in the selected results. The idea is to select the IC function that gave the best results (i.e., picked the best model). The function that is selected will be used for all post analysis.

**Subset Analysis Type** This dropdown box provides three options:

**Complete Enumeration:** This procedure fits the best identified mixture model to all possible subsets of the original data variables. The Best IC scores for all are computed and used to identify the best subset.

**Genetic Algorithm:** When the Best Run file is selected, M<sup>3</sup> will default to this option

if the data used has more than  $p = 6$  variables (63 subsets). This uses the GA to apply the identified best mixture model to all possible subsets of the original data variables, scoring with the Best IC. The GA parameters used in the initial modeling will be used here, though the researcher will be given a chance to change them.

None: Don't perform the subset analysis, this will skip right to influence detection.

A summary file is saved in the same directory as the Best Run file, with either “\_ASUB” (complete enumeration) or “\_GSUB” (genetic algorithm) appended prior to the extension. If the GA is used to perform the subset analysis, the output file is **not** overwritten if it already exists.

Influence Detection: Check this box to perform the influential observation analysis. If subset analysis was performed, this procedure will use the best subset identified. However, the option is given to use any subset of variables desired. The procedure creates a summary file and saves it in the same folder as the Best Run file. This file uses the same name with “\_OUTDET” appended before the extension. A plot is created, but not saved, that shows the IC score for all observations, the group structure from the best model, and identifies influential observations. Lines are drawn to identify the 95% interval.

After all post analysis has been performed, the researcher could load the original data into Matlab, extract the identified best subset (and possibly remove influential observations) and save this new data in the  $M^3$  format.  $M^3$  can then be used to fit just the number of mixtures previously identified in the full dataset, to the subset data.

## • PROGRESS

As  $M^3$  progresses, it will print a lot of output and progress indicators to the Matlab Command Window. Along with this, the bottom panel of the user interface displays several useful progress indicators:

- Current replication - # of Number Replications (or Done),
- Current k - # of Maximum K to Fit,
- Current Information Criterion (or None),
- Minutes required for execution of last: replication (Replic: 0), number of clusters (k: 0), Initialization Method (GARM: 0), Optimization Method / IC (IC: 0),
- Total minutes required - updated before and after each Optimization function call.

## Result Output

There are several files which will be created and saved by  $M^3$  as it progresses; all will go in the `%M3%\“output”\%PStype%\%Input File%_%Maximum K to Fit%` directory. The first file is a Matlab workspace file that stores all the parameters used, the data, and summary results from all replications performed. This file is named following the pattern `“MIXTMR_”%Input File%%Timestamp%.mat`; there are several tables stored in this file worth mentioning:

**GAfil:** This is a character table listing the file saving path and prefix (starting after the `%M3%`) for each replication of the modeling process, for example:

`“\output\Kernel\diabetic_data_6\GKM+GEM_20070225_204142”`

**GAchroms:** This  $(\text{Number Replications} \times [n + 2])$  table holds the best chromosome resulting from each run of the Optimization Function (IC) per replication, by row. The first column records the replication, and the second records the index of the information criterion used. The remainder is, obviously, the chromosome.

**GAcores:** Having one row per IC per replication, the structure of this table is

replication #	IC #	k attempted	best score	percent correct	actual k fit
---------------	------	-------------	------------	-----------------	--------------

The results from the entire process (basically, what’s in this file) are summarized with nice tables and output in a file having the same filename, but ending with `“_SMRY.out”`. For each replication of the modeling process,  $M^3$  will create a human-readable diary of its progress. This is basically what is printed into the Matlab Command Window. All files generated by the same replication will have a filename prefix of

`%Initialization Method%“+”%Optimization Method%%Timestamp%;`

this is part of what is stored in the **GAfil** character matrix.

As previously mentioned, the EM and GEM algorithms produce progress plots that will be saved as Matlab figure files with the suffix `_k_%IC used%_EM.fig` or `_k_%IC used%_GEM.fig` (where  $k$  is the number of clusters attempted).  $M^3$  saves many files per replication. For example, if Maximum K to Fit = 6 and there are 5 information criteria to use with GEM, there will be  $(6 - 1) \times 5 = 25$  GEM progress plots (no optimization needed for  $k = 1$ ). If the Optimization Method is EM, there will be 1 plot per number of mixtures greater than 1 attempted. If the data modeled is of known or simulated bivariate structure,  $M^3$  will generate and save plots showing the actual structure in the left pane and the estimated structure in the right pane for each  $k$  and each information

criterion used. The filename will have a `_k_%IC used%.fig` suffix. The heading of the right pane identifies the Initialization Method, Optimization Method, information criterion used and its score, percent correct, and the actual number of clusters fit. Finally, for each replication, a workspace file is saved, storing all parameters and data used, along with some results. It will be named

`%Initialization Method% "+" %Optimization Method%%Timestamp%.mat;`

the two most important matrices in this workspace are:

**best\_clustassigns:** This is a 3-dimensional matrix that is equivalent to 10

(Number Observations  $\times$  max K to Fit)

matrices – one per possible IC function. This stores the  $\hat{y}_i$  vectors for each  $k$  and IC. For example, **best\_clustassigns**(:,3,2) will hold the class assignments resulting from attempting to fit  $k = 3$  using the second information criterion. If the EM algorithm is used, only **best\_clustassigns**(:,:,1) will contain data (since it doesn't use the IC to optimize).

**SCORES\_CERRS:** This (max K to Fit  $\times 4 \times 10$ ) matrix is structured such that, for each IC and  $k$ , it contains:

final score	percent correct	number clusters fit	number clusters attempted
-------------	-----------------	---------------------	---------------------------

In the case that the EM algorithm was used, **SCORES\_CERRS**(max K to Fit  $\times 4 \times$  number IC) will all be duplicates of **SCORES\_CERRS**(max K to Fit  $\times 4 \times 1$ ).

Along with these two matrices, the variable **rnd\_stat** stores the initial state used for the randomizer (generated at the beginning of each replication, based on the current system time), and **tottim** has the total number of minutes that the replication required.

The M<sup>3</sup> toolbox will be incorporated into the *Information Complexity Toolbox for MATLAB* currently under development by Andrew and Dr. Bozdogan. Upon email request (ahowe42@gmail.com or bozdogan@utk.edu), the software may be made available for noncommercial use.

## Appendix 2: Datasets

All simulated datasets are generated using the multivariate power exponential distribution.

### Simulation S1 - Mixed Overlapping

Table A.1: Simulation S1 - Data Generation Parameters.

$k$	$\pi_k$	$\beta_k$	$\mu_k$	$\Sigma_k$
1	0.33	1.5	$\begin{bmatrix} 0.00 \\ -1.00 \end{bmatrix}$	$\begin{bmatrix} 10.00 & 0.50 \\ 0.50 & 2.00 \end{bmatrix}$
2	0.33	1	$\begin{bmatrix} -3.00 \\ 3.00 \end{bmatrix}$	$\begin{bmatrix} 2.00 & 2.00 \\ 2.00 & 10.00 \end{bmatrix}$
3	0.33	1	$\begin{bmatrix} 6.00 \\ 3.00 \end{bmatrix}$	$\begin{bmatrix} 3.10 & 2.00 \\ 2.00 & 4.45 \end{bmatrix}$

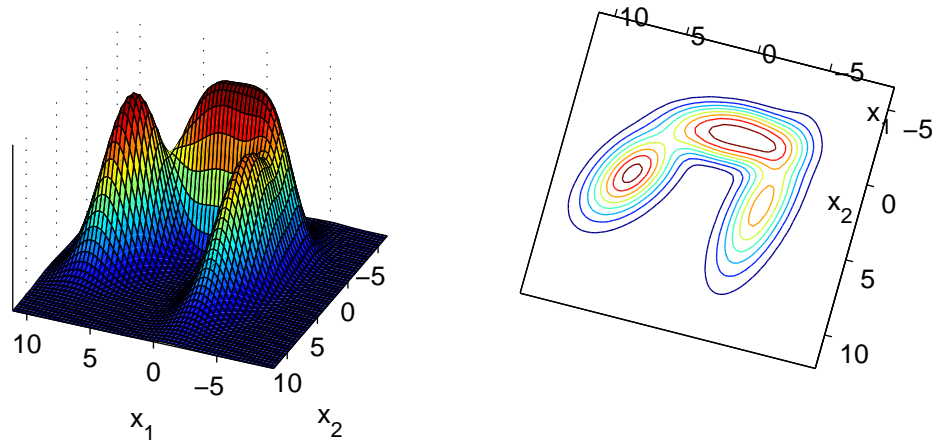


Figure A.3: Simulation S1 - Surface and Contour Plots.

## Simulation S2 - Ellipsoidal Overlapping

Table A.2: Simulation S2 - Data Generation Parameters.

$k$	$\pi_k$	$\beta_k$	$\mu_k$	$\Sigma_k$
1	0.30	1	$\begin{bmatrix} 0.63 \\ 0.97 \end{bmatrix}$	$\begin{bmatrix} 1.17 & 0.33 \\ 0.33 & 0.12 \end{bmatrix}$
2	0.50	1	$\begin{bmatrix} 1.01 \\ 0.59 \end{bmatrix}$	$\begin{bmatrix} 0.34 & -0.24 \\ -0.24 & 0.18 \end{bmatrix}$
3	0.20	1	$\begin{bmatrix} 0.32 \\ -0.51 \end{bmatrix}$	$\begin{bmatrix} 0.14 & 0.05 \\ 0.05 & 0.03 \end{bmatrix}$

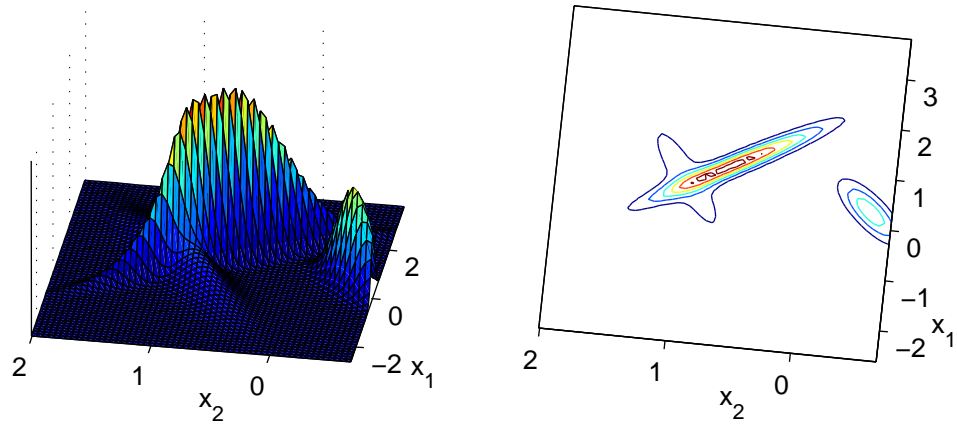


Figure A.4: Simulation S2 - Surface and Contour Plots.

## Simulation S3 - Spherical Overlapping

Table A.3: Simulation S3 - Data Generation Parameters.

$k$	$\pi_k$	$\beta_k$	$\mu_k$	$\Sigma_k$
1	0.33	1	$\begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$	$\begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 2.00 \end{bmatrix}$
2	0.33	1	$\begin{bmatrix} 8.30 \\ 8.10 \end{bmatrix}$	$\begin{bmatrix} 4.00 & 0.00 \\ 0.00 & 4.00 \end{bmatrix}$
3	0.33	1	$\begin{bmatrix} 5.00 \\ 8.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 & 0.00 \\ 0.00 & 0.50 \end{bmatrix}$

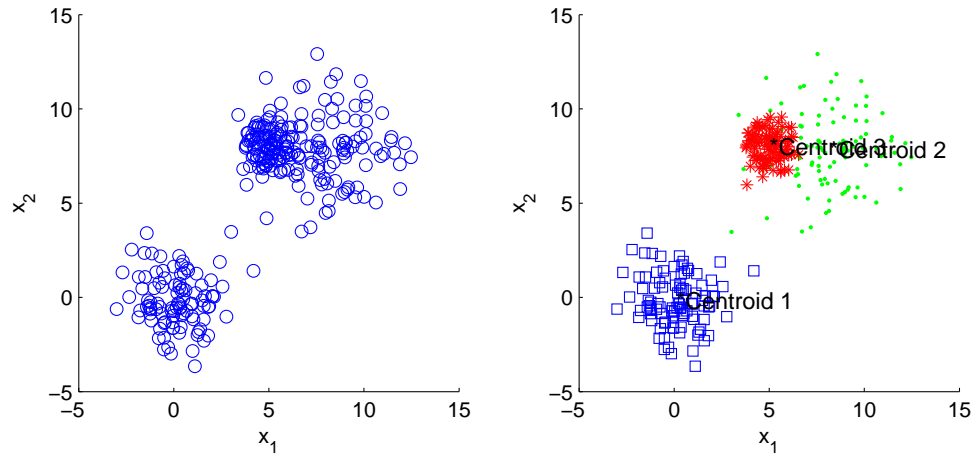


Figure A.5: Simulation S3 - Sample Scatter Plot of  $X_1$  Against  $X_2$ .

## Simulation S4 - Nonoverlapping

Table A.4: Simulation S4 - Data Generation Parameters.

$k$	$\pi_k$	$\beta_k$	$\mu_k$	$\Sigma_k$
1	0.33	1	$\begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$	$\begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 0.50 \end{bmatrix}$
2	0.33	1	$\begin{bmatrix} 8.30 \\ 8.10 \end{bmatrix}$	$\begin{bmatrix} 4.44 & 2.17 \\ 2.17 & 6.18 \end{bmatrix}$
3	0.33	1	$\begin{bmatrix} -5.00 \\ 10.00 \end{bmatrix}$	$\begin{bmatrix} 3.10 & 2.00 \\ 2.00 & 4.45 \end{bmatrix}$

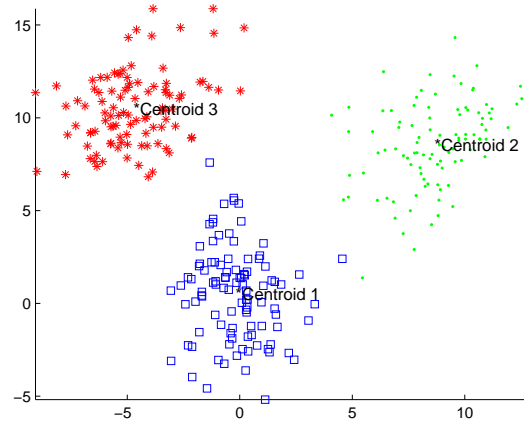


Figure A.6: Simulation S4 - Sample Scatter Plot of  $X_1$  Against  $X_2$ .



## Simulation S5 - Spherical Overlapping

Table A.5: Simulation S5 - Data Generation Parameters.

$k$	$\pi_k$	$\beta_k$	$\mu_k$	$\Sigma_k$
1	0.30	0.50	$\begin{bmatrix} -2.00 \\ -2.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 & -0.10 \\ -0.10 & 0.50 \end{bmatrix}$
2	0.40	2.00	$\begin{bmatrix} 3.00 \\ 3.00 \end{bmatrix}$	$\begin{bmatrix} 2.00 & 0.50 \\ 0.50 & 0.25 \end{bmatrix}$
3	0.30	1.00	$\begin{bmatrix} -5.00 \\ 3.00 \end{bmatrix}$	$\begin{bmatrix} 0.50 & 0.00 \\ 0.00 & 3.00 \end{bmatrix}$

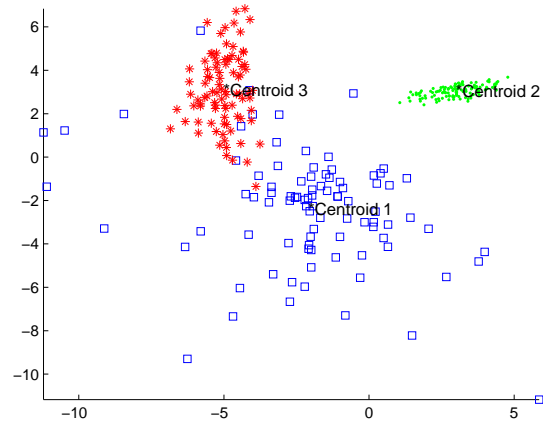


Figure A.7: Simulation S5 - Sample Scatter Plot of  $X_1$  Against  $X_2$ .

## Real Data - Aorta

The *Nuclear Magnetic Resonance* aorta data analyzed here was collected by Pearlman (1986) at the Medical School of the University of Virginia. There are observations from  $n = 418$  patients on 16 different image acquisition variables. Including direction and orientation variables, we have  $p = 20$  variables. The first  $n_1 = 194$  patients exhibited *early atheroma*, and the remaining  $n_2 = 224$  patients were *healthy*. In Figures A.8 and A.9, we see market nonnormality as well as distinct patterns of good and poor class separation.

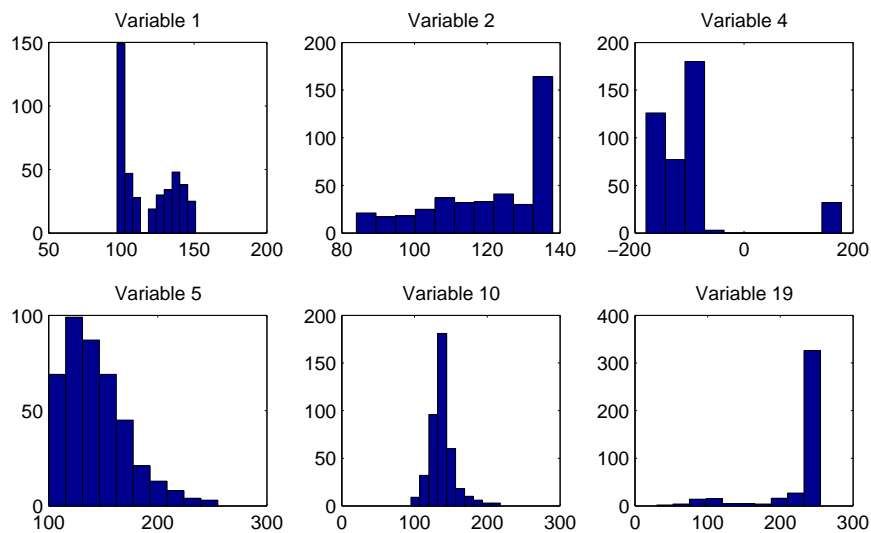
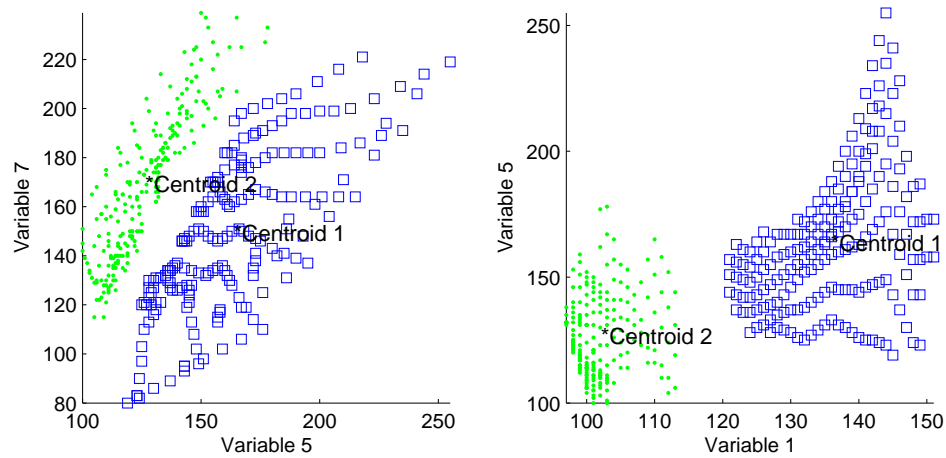
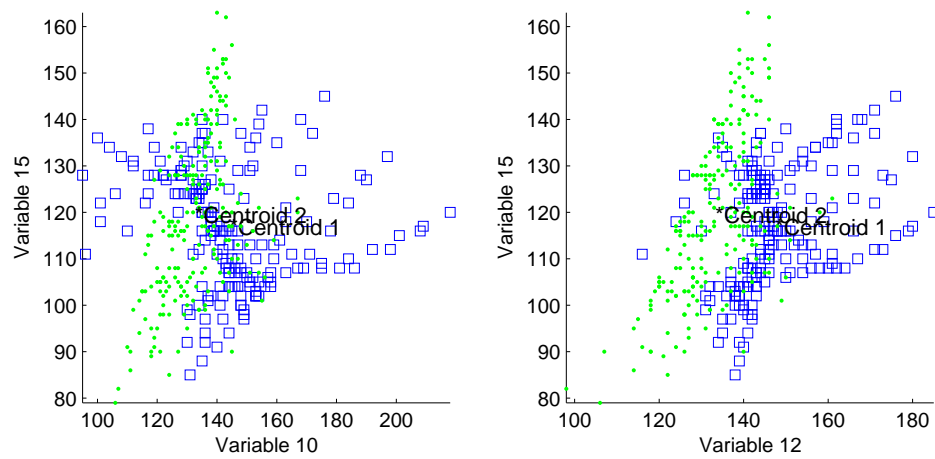


Figure A.8: Aorta Data - Demonstrating Nonnormal Characteristics.



(a) Demonstrating Good Separation Between 2 Pairs of Dimensions.



(b) Demonstrating Poor Separations Between 2 Pairs of Dimensions.

Figure A.9: Aorta Data Bivariate Scatter Plots.

## Real Data - Cancer

Next, we have a medical dataset first used in Street et al. (1993), from the UCI data repository. Features are computed from a digitized image of a fine needle aspirate (biopsy) of a breast mass from  $n = 569$  patients. They describe characteristics of the cell nuclei present in the image. Ten real-valued features are computed for each cell nucleus, with the plotmatrix of the first five shown in Figure A.10.

- a) radius (mean of distances from center to points on the perimeter) -  $[x_1, x_{11}, x_{21}]$
- b) texture ( $\sigma$  of gray-scale values) -  $[x_2, x_{12}, x_{22}]$
- c) perimeter -  $[x_3, x_{13}, x_{23}]$
- d) area -  $[x_4, x_{14}, x_{24}]$
- e) smoothness (local variation in radius lengths) -  $[x_5, x_{15}, x_{25}]$
- f) compactness ( $\frac{\text{perimeter}^2}{\text{area}} - 1.0$ ) -  $[x_6, x_{16}, x_{26}]$
- g) concavity (severity of concave portions of the contour) -  $[x_7, x_{17}, x_{27}]$
- h) concave points (number of concave portions of the contour) -  $[x_8, x_{18}, x_{28}]$
- i) symmetry -  $[x_9, x_{19}, x_{29}]$
- j) fractal dimension ("coastline approximation" - 1.0) -  $[x_{10}, x_{20}, x_{30}]$

The mean, standard error, and mean of the three largest values of these features "worst" were computed for each image, resulting in  $p = 30$  features. Thus, variables 1 – 3 relate to the radius, 4 – 6 relate to the texture, .... This data is composed of  $K = 2$  groups;  $n_1 = 212$  patients had malignant tumors, while the masses of the other  $n_2 = 357$  were benign (noncancerous).

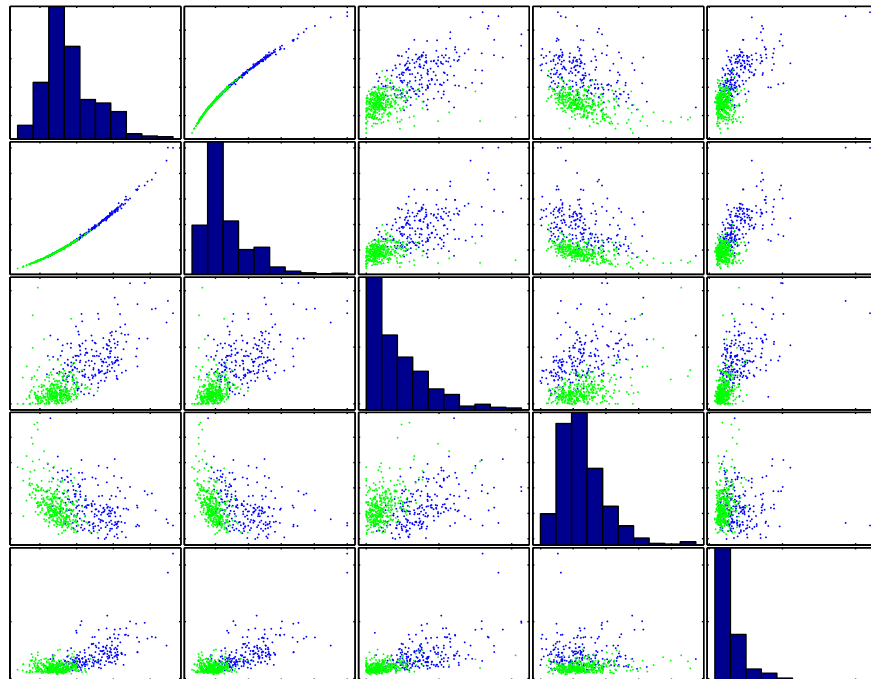


Figure A.10: Cancer data - Scatter Plot Matrix of Variables a) Through e).

## Real data - Colon

This dataset, provided by Dr. Bozdogan, is composed of  $n = 65$  observations from  $K = 5$  groups. The 2<sup>nd</sup> and 3<sup>rd</sup> groups have 17 observations each. The first group has 13, the fourth has 7, and the 5<sup>th</sup> has the remaining 11. Most of the  $p = 5$  variables have very low correlations with each other. Figure A.11 demonstrates the nonnormality and extreme overlap exhibited in this data.

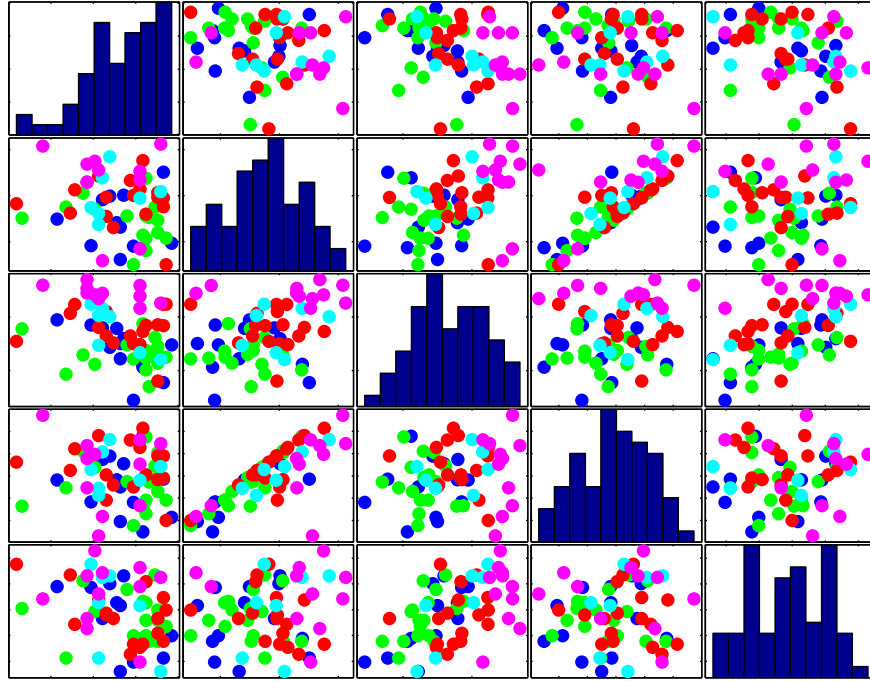


Figure A.11: Colon data - Grouped Scatterplot Matrix.

## Real Data - Diabetic

This dataset is composed of  $K = 3$  types of patients from a diabetes study of Andrews and Herzberg (1985). Five medical measurements relating to insulin usage were taken on  $n_1 = 33$  *overt diabetic*,  $n_2 = 36$  *chemical diabetic*, and  $n_3 = 76$  *non-diabetic* patients:

$x_1$  = Relative Weight

$x_2$  = Fasting Plasma Glucose

$x_3$  = Glucose Area

$x_4$  = Insulin Area

$x_5$  = SSPG

The parallel coordinates plot (Wegman, 1986) in Figure A.12 shows the clear overlap among the groups, especially for  $x_4$  and  $x_5$ , as well as the nice separation in the *Glucose Area* measurements. In fact, due to the clinical similarity between *non-diabetic* and *chemical diabetic* patients, finding either  $\hat{K} = 2$  or  $\hat{K} = 3$  is acceptable for this dataset. Please see Figure A.13 for the scatter plot matrix of this data.

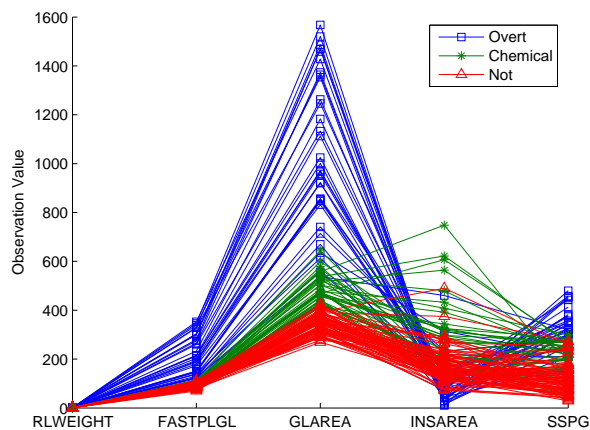


Figure A.12: Diabetic Data - Parallel Coordinates Plot.

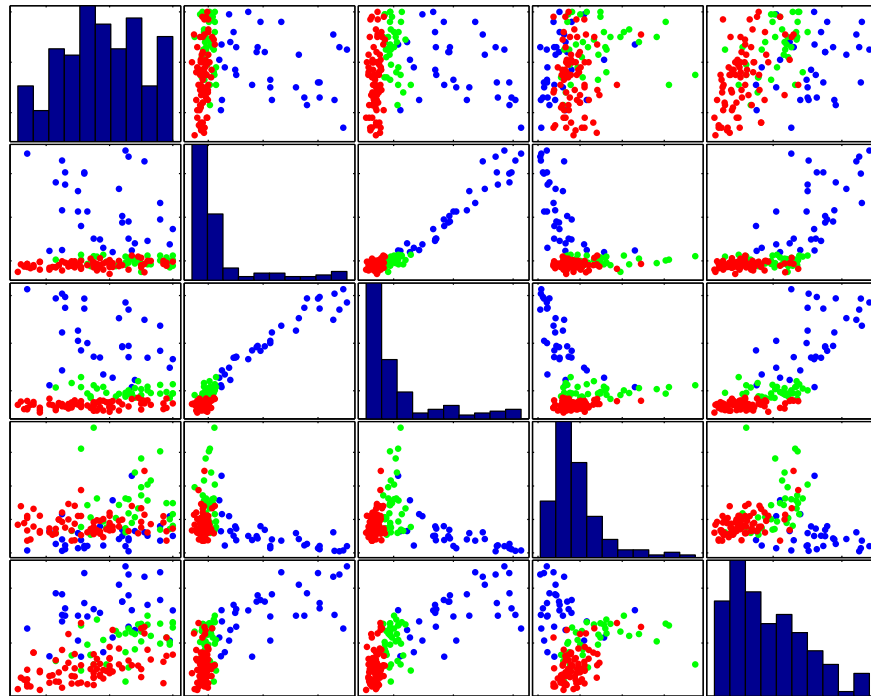


Figure A.13: Diabetic Data - Scatter Plot Matrix.



## Real Data - Iris

This dataset consists of  $n = 150$  observations of  $p = 4$  flower characteristics: *petal length*, *petal width*, *sepal length*, and *sepal width*. The iris data contains  $K = 3$  groups; 50 observations each from the varieties *Iris Setosa*, *Iris Versicolor*, and *Iris Virginica*.

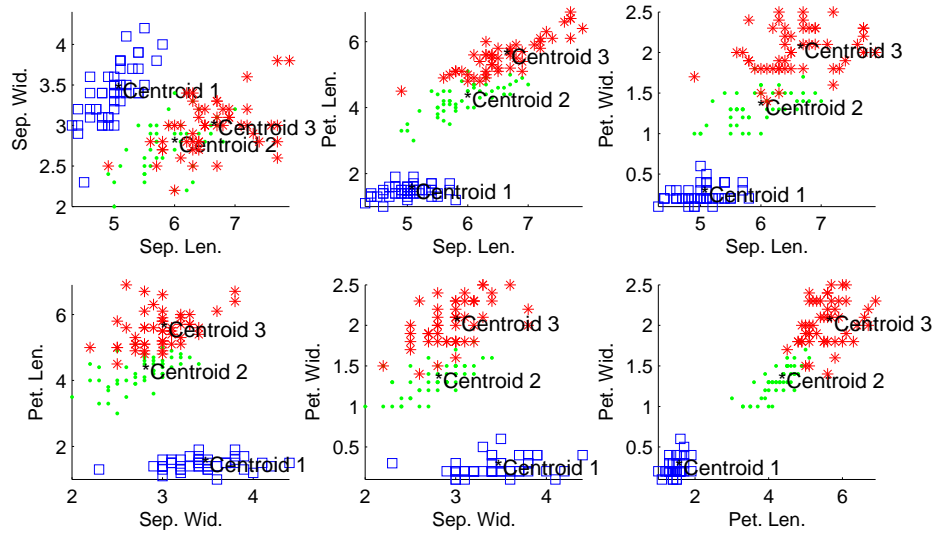


Figure A.14: Iris Data - Pairwise Scatter Plots.

## Real data - Wine

This is the wine recognition dataset of *Fiorina, M. et al*, used in Aeberhard et al. (1992). These data are the results of a chemical analysis of  $n = 178$  wines grown in the same region in Italy but derived from  $K = 3$  different cultivars ( $n_1 = 59, n_2 = 71, n_3 = 48$ ). The analysis determined the quantities of  $p = 13$  constituents found in each of the three types of wines. The variables are shown in Table A.6. The mixing proportions for the three groups are 33.15%, 39.89%, 26.97%.

Table A.6: Wine data - Variables.

Variable		Variable	
$x_1$	Alcohol	$x_8$	Non-flavonoid Phenols
$x_2$	Malic Acid	$x_9$	Proanthocyanins
$x_3$	Ash	$x_{10}$	Color Intensity
$x_4$	Alcalinity of Ash	$x_{11}$	Hue
$x_5$	Magnesium	$x_{12}$	OD280/OD315 of Diluted Wines
$x_6$	Total Phenols	$x_{13}$	Proline
$x_7$	Total Flavonoids		

Figure A.15 and A.16 show scatter plot matrices for this dataset, displaying substantial overlap of the three groups, and nonnormal shapes in many dimensions.

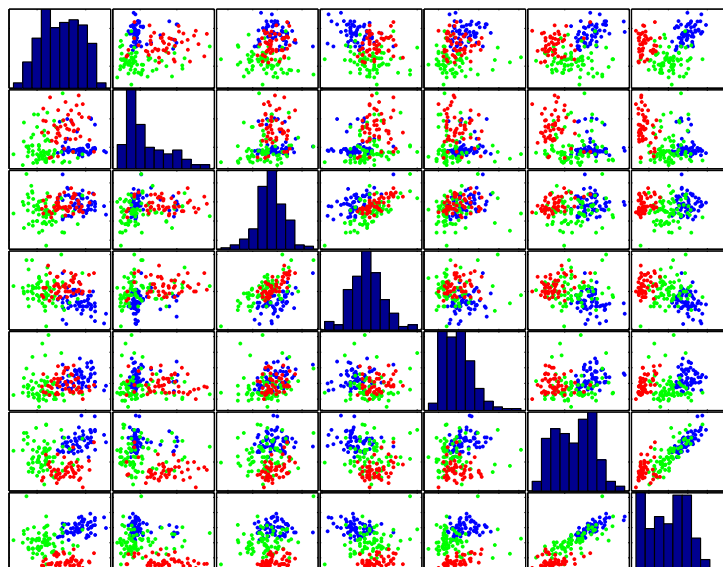


Figure A.15: Wine data - Grouped Scatterplot Matrix for  $x_1 \dots x_7$ .

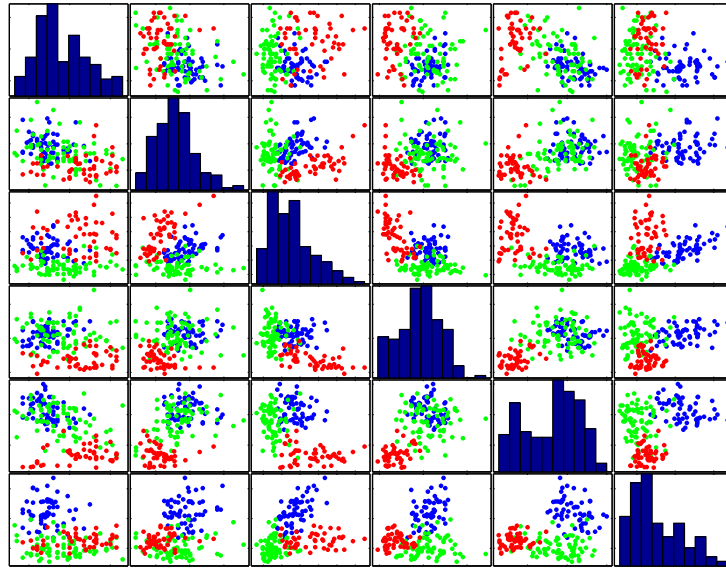


Figure A.16: Wine data - Grouped Scatterplot Matrix for  $x_8 \dots x_{13}$ .

# Vita



*“I live to learn so I can learn to live.”* - J. Andrew Howe

J. Andrew Howe was born in California in 1975, a year that will long be remembered for the theatrical release of Mel Brook’s *Young Frankenstein*. In 1993, Andrew matriculated to California Baptist College. Four years later, he graduated *Cum Laude*, having obtained his Bachelor’s degree in Pure and Applied Mathematics, along with a minor in physics. While working full time, he attended evening school, receiving his MBA with a concentration in finance from Keller Graduate School of Management in 2001.

Along the way, Andrew acquired invaluable information in both the financial and healthcare industries. At Golden Gate Financial Group, he was introduced to the art and science of developing profitable trading models. He was solely responsible for developing a volatility-based model allocation and arbitration system that, in back-testing, increased returns and decreased volatility. It was here that Andrew first met Dr. Hamparsum Bozdogan, who suggested he should come to Tennessee and study under him.

Several years after this “chance” encounter, Andrew joined the department of Statistics, Operations and Management Science at the University of Tennessee, Knoxville in 2005. In 2007, he graduated with a Master’s degree in Statistics, having spent his summers performing research on a Fellowship with Dr. Bozdogan. Andrew intends to complete the requirements for the PhD in Statistics by

spring of 2009. Andrew currently works in the *Power Supply & Fuels* organization within Tennessee Valley Authority in the Chattanooga office. He has been happy with the opportunity to apply his skills in this dynamic arena.

Andrew is passionate about learning and reading, science, financial trading, learning, cutting-edge research, the human brain, classical music, and did I say learning? His research interests include Multivariate Modeling, Information Complexity, Model Selection, Robust Modeling, and Statistical Computing.