5-2006

Applications of Modern Statistical Methods to Analysis of Data in Physical Science

James Eric Wicker
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a dissertation written by James Eric Wicker entitled "Applications of Modern Statistical Methods to Analysis of Data in Physical Science." 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 Physics.

William E. Blass, Major Professor

We have read this dissertation and recommend its acceptance:

Halima Bensmail, Hamparsum Bozdogan, Marianne Breinig, Chia C. Shih

Accepted for the Council:

Carolyn R. Hodges

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 James Eric Wicker entitled "Applications of Modern Statistical Methods to Analysis of Data in Physical Science." 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 Physics.

William E. Blass
Major Professor

We have read this dissertation
and recommend its acceptance:

Halima Bensmail

Hamparsum Bozdogan

Marianne Breinig

Chia C. Shih

Accepted for the Council:

Anne Mayhew
Vice Chancellor and Dean of
Graduate Studies

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

Applications of Modern Statistical Methods to Analysis of Data in Physical Science

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

James Eric Wicker
May 2006

Abstract

Modern methods of statistical and computational analysis offer solutions to dilemmas confronting researchers in physical science. Although the ideas behind modern statistical and computational analysis methods were originally introduced in the 1970's, most scientists still rely on methods written during the early era of computing. These researchers, who analyze increasingly voluminous and multivariate data sets, need modern analysis methods to extract the best results from their studies.

The first section of this work showcases applications of modern linear regression. Since the 1960's, many researchers in spectroscopy have used classical stepwise regression techniques to derive molecular constants. However, problems with thresholds of entry and exit for model variables plagues this analysis method. Other criticisms of this kind of stepwise procedure include its inefficient searching method, the order in which variables enter or leave the model and problems with overfitting data. We implement an information scoring technique that overcomes the assumptions inherent in the stepwise regression process to calculate molecular model parameters. We believe that this kind of information based model evaluation can be applied to more general analysis situations in physical science.

The second section proposes new methods of multivariate cluster analysis. The K-means algorithm and the EM algorithm, introduced in the 1960's and 1970's respectively, formed the basis of multivariate cluster analysis methodology for many years. However, several shortcomings of these methods include strong dependence on initial seed values and inaccurate results when the data seriously depart from hypersphericity. We propose new cluster analysis methods based on genetic algorithms that overcomes the strong dependence on initial seed values. In addition, we propose a generalization of the Genetic K-means algorithm which can accurately identify clusters with complex hyperellipsoidal covariance structures. We then use this new algorithm in a genetic algorithm based Expectation-Maximization process that can accurately calculate parameters describing complex clusters in a mixture model routine. Using the accuracy of this GEM algorithm, we assign information scores to cluster calculations in order to best identify the number of mixture components in a multivariate data set. We will showcase how these algorithms can be used to process multivariate data from astronomical observations.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recent decades witnessed the emergence of modern methods in statistical analysis and a dramatic increase in computational power. At the same time, almost every field of science and engineering experienced an explosion in the amount of data collected from experiments and observational studies. In many disciplines, the rate at which data is being collected and warehoused continues to accelerate. In astronomy, for instance, due to improvements in detector and storage technology, vast amounts of astronomical data are being archived each year. Szalay (2002) writes that astronomers will have to surmount terabytes, and soon petabytes, of data. Other experts point out that there are currently more than 100 terabytes archived in astronomical databases, with the amount continuously growing. By comparison, the size of the human genome is about 1 gigabyte, and size of the Library of Congress is about 20 terabytes (Babu and Djorgovski 2004).

New methods of analysis have been developed to deal with this onslaught of data (Bozdogan 2004). This dissertation will explore how some of these modern statistical and computational methods can provide insight into the analysis of physical

data. This is an interdisciplinary endeavor, drawing on topics from physical science, statistics, and computational methods.

This work is divided into two sections. The first section addresses modern multivariate regression techniques. Multivariate regression is a data processing method that attempts to find the best equation that describes how some observed independent variables are related to one or more dependent variables. Although regression methods currently used by most researchers utilize stepwise analysis, this method suffers some fundamental handicaps. Stepwise regression analysis has little basis in statistical theory. It also requires that the user specify thresholds for classical F tests to calculate the significance of variables. This approach to regression generally yields models that generalize poorly. Moreover, the order in which variables are processed in a stepwise procedure generally affects the computed model. Even for the same data set, different orders of entry and exit of the variables can yield significantly different final models. The compounded effect of these problems can cast doubt on the reliability of subsequent physical interpretations.

Regression analysis based on information theory offers solutions that overcome the inherent shortcomings of classical regression. Instead of relying on arbitrary F test thresholds, information scores are calculated for different combinations of parameters. These scoring functions try to find the most parsimonious model that best describes the system under study. Moreover, unlike the ad hoc stepwise method, regression analysis based on information scores is well grounded in statistical theory. The benefits of information based regression become especially important with increasing numbers of variables. Since its introduction in the early 1970's until now, information based statistical analysis has been applied in economics and social science, but has remained mostly unknown to researchers in physical science. This is

especially true of regression using ICOMP as a scoring function (Bozdogan 2004). Because of its relatively recent introduction, multivariate regression scored with ICOMP has not been applied to analysis of physical data until this work.

Chapters 2 through 7 of this dissertation introduce the benefits of using multivariate regression in physical science. Chapter 2 reviews methods of classical multivariate regression, while Chapter 3 shows the development of information based multivariate regression. As the number of possible variables increases, computational shortcuts may be necessary to effectively reap the benefits of multivariate regression with information scores. Chapter 4 shows how binary genetic algorithms can be implemented in a multivariate regression situation to deal with a large number of possible variables. The power series representation of the vibration-rotation Hamiltonian (Blass and Nielsen 1974) forces researchers to consider different orders of magnitude when applying multivariate regression analysis to the analysis of molecular spectra. Chapter 5 reviews the development of the theory of molecular spectra. Chapter 6 then introduces a new method of implementing information scored regression when analyzing data from an expanded power series model. Chapter 7 showcases examples of this new method in the analysis of molecular spectra.

The second section of this work is about methods of cluster analysis. Cluster analysis classifies data into categories according to some rule or property. Processing ever-growing data sets necessitates new methods to handle such data. Researchers using cluster analysis have historically relied on seed based methods to calculate cluster partitions. The results of seed based cluster algorithms depend strongly on initial values. Traditionally, the K-means algorithm has supplied the basis of these cluster calculations. The K-means method computes clusters based on their Euclidean distance from seed values, and iteratively recalculates cluster memberships and centroids until no change occurs. Krishna and Murty (1999) proposed a method of overcoming

dependence on initial seed values when they introduced the Genetic K-means algorithm. The Genetic K-means algorithm represents cluster assignments as strings of integers. The population of these strings undergoes genetic operations, searching for the minimum value in cluster variance.

Although the K-means algorithm partitions the data into a given number of clusters, it can only accurately partition hyperspherical data whose clusters do not overlap. For data that have hyperellisoidal structure that may overlap, researchers must implement more advanced methods. The Expectation-Maximization (EM) algorithm (McLachlan and Krishnan 1997) tries to maximize the posterior probabilities of group membership in clustered data. The EM algorithm calculates the mixture of finite distributions model of the clustered data by computing the maximum likelihood estimates of parameters describing the components. The traditional EM algorithm uses a K-means initialization to calculate initial cluster assignments and cluster parameter estimations. It then iteratively recalculates cluster probabilities and parameter estimates using a gradient ascent method. Bozdogan (1994) proposed combining the EM algorithm with information scoring to identify the best number of clusters present in a multivariate data set. This information scoring method depends heavily on how accurately the algorithm models cluster means and covariances.

In order to overcome the strong dependence on initialization of the traditional EM algorithm, we propose a new Genetic Expectation Algorithm (GEM). This algorithm can accurately model cluster parameters without relying heavily on initialization. We will show how this algorithm can accurately estimate cluster parameter values, even in situations where clusters overlap and depart strongly from hypersphericity.

Chapters 8 through 11 of this dissertation propose new methods of cluster analysis. We show how the Genetic K-means algorithm (Krishna and Murty 1999) representation can be used in an Expectation-Maximization context. Chapter 8 will show how

the Genetic K-means algorithm can be generalized to hyperellisoidal data. Chapter 9 proposes the multivariate EM algorithm for clustering which uses GA strings, while Chapter 10 implements this new method in an information scoring routine that can identify the best number of clusters in a multivariate data set. Chapter 11 demonstrates how these Genetic Expectation Maximization (GEM) methods can process astronomical data. The dissertation will conclude with Chapter 12.

Some readers may have the impression that this dissertation contains some lengthy discussions of statistical theory. They might also observe that some theoretical developments and algorithmic descriptions are repeated in multiple chapters. The reasons this author elected to structure the material in this way are two-fold. The first is that, currently, there are few widely-published works that describe the advantages of using modern data analysis methods, especially as applied to physical science. We hope that this dissertation can introduce the benefits of these new methods to a wider audience of researchers. The second is that, if the reader is mainly interested in applications of the methods, the material in those sections is self-contained. These readers can treat the theoretical sections lightly and regard the application chapters as independent works.

Any interested reader that has questions or comments about this work can contact the author at jewicker@gmail.com.

Chapter 2

Classical Linear Regression

2.1 Introduction

Linear regression is a data analysis method that explores linear relationships between measurable or observable quantities. Montgomery et al. (2001, page 1) states the definition of regression analysis to be as follows: "Regression analysis is a statistical technique for investigating and modeling the relationship between variables. Applications of regression are numerous and occur in almost every field, including engineering, the physical and chemical sciences, ... In fact, regression may be the most widely used statistical technique."

According to the Montgomery definition of regression, there are two main motivations for using regression analysis. The first is where researchers try to decide if there is a relationship between one or more independent variables and at least one dependent variable. After they have satisfied the first goal, they can turn to the second motivation, which is to decide which variables best describe the observed relationship. One example of this case was in material science, where researchers had over 100 possible variables that described a material under study. These researchers

were interested in which subset of variables were best to describe the material under study. From the large number of possible variables, they found a subset to include in the model, which helped guide their future study of this material (Bozdogan, private communication).

We can again reference Montgomery for the definition of a simple linear regression model (Montgomery et al. 2001, page 13): "...the simple linear regression model, that is, a model with a single regressor x that has a relationship with a response y that is a straight line. This linear regression model is

$$y = \beta_0 + \beta_1 x + \epsilon \tag{2.1}$$

where the intercept β_0 and the slope β_1 are unknown constants and ϵ is a random error component." In most regression modeling situations, the random error ϵ has an average value of zero. Montgomery also gives a description of the independent and dependent variables as (Montgomery et al. 2001, page 3): "Customarily x is called the independent variable and y is called the dependent variable. In case this causes confusion with the concept of statistical independence, we can also refer to x as the predictor or regressor variable, and y as the response variable."

Often times, there are many possible measurable or observable quantities that may be related to a response. Deciding the best variables to include in the regression equation leads to multiple linear regression. According to Montgomery et al. (2001, page 6): "In general, the response variable may be related to k regressor variables, $x_1, x_2, \ldots, x_k$, so that

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \ldots + \beta_k x_k + \epsilon \tag{2.2}$$

This is called a multiple linear regression model because more than one regressor is involved. The adjective linear is employed to indicate that the model is linear in the parameters $\beta_0, \beta_1, \beta_2, \ldots \beta_k$, not because y is a linear function of x."

Researchers can use regression analysis to connect experimental data with theoretical model equations. We can again reference the textbook description of this process (Montgomery et al. 2001, page 6): "An important application of regression analysis is to estimate the unknown parameters in the regression model. This process is also called fitting the model to the data." If theoretical equations can be written as linear equations in the parameters, then the regression constants become parameter estimates derived from experimental data. We must be mindful that, if the researcher is engaged in exploratory data analysis, the first motivation described by Montgomery et al. (2001), then the regression coefficients do not necessarily correspond with a physical model, so it may be misleading to call them "parameters." They are parameters in the sense that they define slopes in some high dimensional measurement space, but unless these equations are in the form of physical laws, the "parameters" do not correspond to physical quantities. Only if the equations follow the form of a physical law do the calculated regression parameters correspond to physical quantities. An example where the "model parameters" correspond to physical quantities is the equation describing the acceleration of gravity $y = \frac{1}{2}gx^2$, where x^2 is the regressor variable and y is the response variable, and $\frac{1}{2}g$ is the coefficient to the calculated.

The ideas behind linear regression started with the investigation of relationships in astronomy (Babu and Djorgovski 2004), and linear regression has enjoyed wide application in physical science. An example involves the analysis of Hubble's redshift data. Plotting the observed redshift of a galaxy against its distance revealed a linear relationship indicating that the farther the distance between galaxies, the faster they are receding from each other (see figure 2.1) (Hubble 1929).

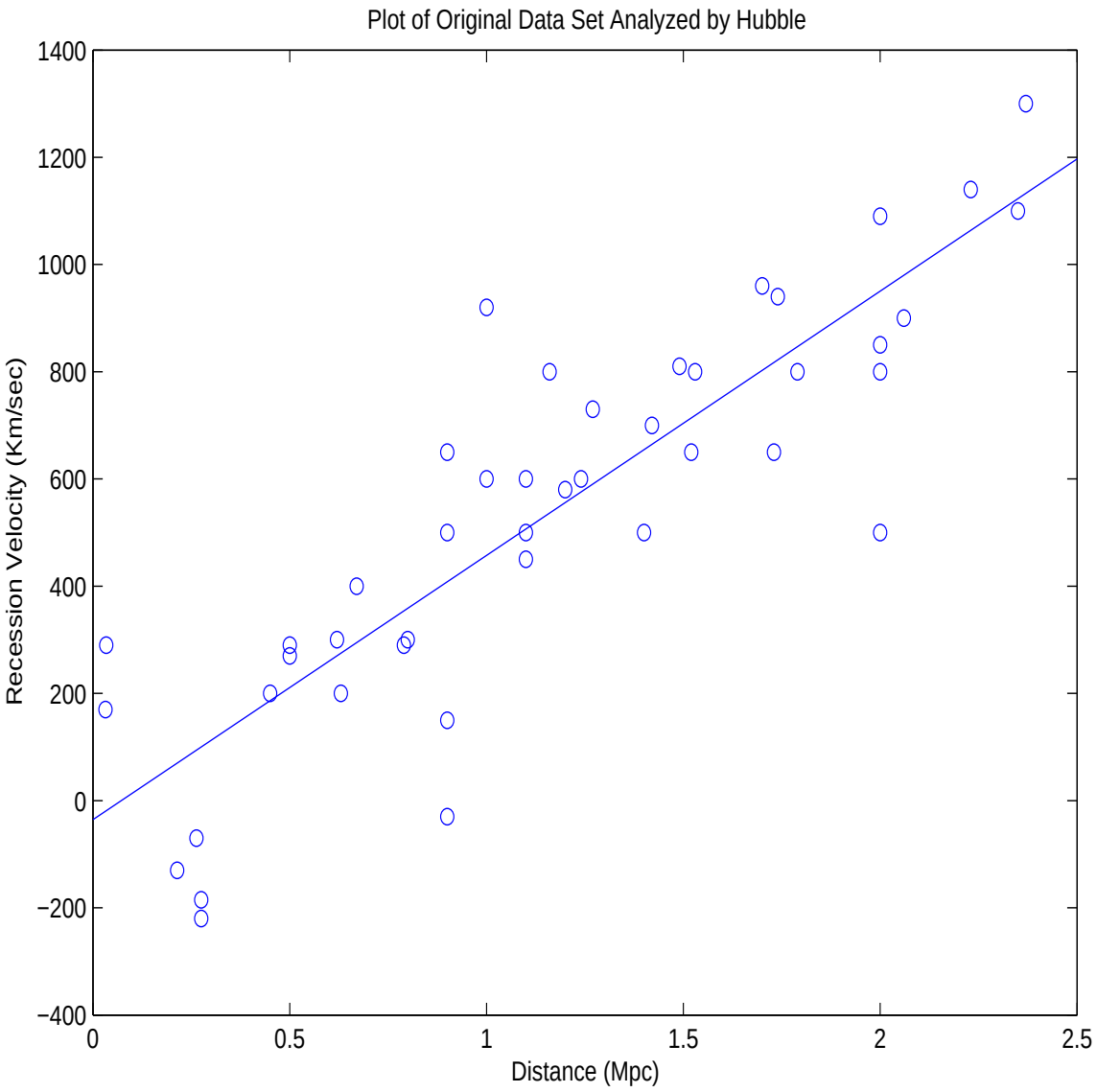

Figure 2.1: Hubble's Distance Verses Velocity Plot.

Hubble used regression analysis to derive the relationship that we know today as Hubble's law.

$$V = H_0 D \tag{2.3}$$

The inverse of H_0 (usually with units of years), is the age of the Universe.

Much of the data collected and stored from modern science and engineering studies are multivariate. The data compiled by Zhang and Zhao from sources like the US Naval Observatory and the Sloan Digital Sky Survey has 10 variables, and a similar archive has 109 measured quantities (Zhang and Zhao 2003). This can be an example of exploratory data analysis, where deciding which quantities best describe a process is an empirical exercise, but the calculated slope "parameters" do not necessarily correspond to physical quantities. Trying to find linear relationships between these observed measurements not only requires slope and intercept calculations, but also variable selection methods offered by linear regression methodology.

Methods of deciding which variables are most important is a contentious debate between the different statistical schools-of-thought. Classical statistics, which has historically been used by researchers in physical science, emphasizes a frequentist interpretation of data analysis. Frequentist interpretation regards probabilities to be average long-run limits of repeated experiments, and performs hypothesis tests to make decisions about data results. Other major factions of modelers include the *Bayesian* school and the *Information theory* school. The Bayesian school considers prior knowledge to be important while the Information theory school draws from both the classical and Bayesian perspectives, and is considered to be the most modern form of statistical modeling.

The debate about which interpretation of statistics is best has started to diffuse into other areas of science. In his article "Why isn't every physicist a Bayesian," Cousins (1995) describes how the underlying assumptions of data analysis can lead

to different results for the same data set. Cousins points out that most researchers in his field of particle physics rely on frequentist data analysis methodology, but that Bayesian methodology has been making inroads into published work. In his abstract, he states, "... many other particle physicists may frequently think in a Bayesian manner without realizing it." He goes on to show how different data analysis methods can lead to different results, later stating that "...some of the most common analysis problems in particle physics go straight to the core of the classical Bayesian debate in a way that cannot be avoided."

Methods of statistical modeling allow a researcher to test variables that the researcher hypothesizes may be related to observed responses. Many texts have been written that describe statistical modeling methodologies, including Draper and Smith (1966) and Montgomery et al. (2001). The goal of statistical modeling is to produce an equation that best summarizes the observed data and can be used to make predictions about future observations. The model attempts to reach a compromise between including enough variables so that the observed phenomenon can be adequately described and overfitting the model by including unnecessary variables. Overfitting not only complicates the physical interpretation of the data, but also may include correlated variables that exaggerate error estimates of the study.

In order to understand the contrasts between the different statistical modeling methodologies, we first need to review classical linear regression. Classical regression methods rely on hypothesis testing and minimizing the error variances. This chapter shall outline classical methods of statistical modeling using linear least-squares analysis. We will concentrate on parametric statistical modeling, where the observed variables are assumed to arise from analytic probability distributions whose parameters can be estimated. Parametric modeling methods start by assuming the data

follow some probability distribution, and then derives methods to estimate parameters based on this distribution. This contrasts with non-parametric methods that do not assume a probability distribution, but rather use other numerical properties to estimate parameters. This chapter will show the theoretical development of the least-squares estimators, methods of testing statistical significance in a model, and a discussion of modeling assumptions associated with linear least-squares.

2.2 Linear Least Squares

Classical linear statistical modeling is based on least-squares normal equations. These equations are derived such that the sum of squared differences between observations and a straight line in measurement space is minimized and the parameters of the line are estimated. We will follow the convention that a hat over a symbol denotes the estimator of that quantity. Let us assume that we have k regressor variables and one response variable. We would like to calculate a linear least-squares equation such as

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_i \end{aligned} \tag{2.4}$$

The least-squares function is

$$\begin{aligned} S(\beta_0, \beta_1, \ldots, \beta_k) &= \sum_{i=1}^{n} \varepsilon_i^2 \\ &= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2 \end{aligned} \tag{2.5}$$

The least-squares function must be minimized with respect to the β_i regression coefficients, so the equations must satisfy the following conditions:

$$\begin{aligned}
\frac{\partial S}{\partial \beta_0}_{\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_k} &= -2\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{k}\hat{\beta}_j x_{ij}\right) = 0 \\
\frac{\partial S}{\partial \beta_j}_{\hat{\beta}_0,\hat{\beta}_1,\ldots,\hat{\beta}_k} &= -2\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{k}\hat{\beta}_j x_{ij}\right) x_{ij} = 0, \quad j = 1,\ldots,k
\end{aligned} \tag{2.6}$$

Simplifying these equations leads to the least-squares normal equations

$$\begin{aligned}
n\hat{\beta}_0 + \hat{\beta}_1\sum_{j=1}^{n} x_{i1} + \cdots + \hat{\beta}_k\sum_{j=1}^{n} x_{ik} &= \sum_{i=1}^{n} y_i \\
\hat{\beta}_0\sum_{i=1}^{n} x_{i1} + \hat{\beta}_1\sum_{j=1}^{n} x_{i1}^2 + \cdots + \hat{\beta}_k\sum_{j=1}^{n} x_{i1}x_{ik} &= \sum_{i=1}^{n} x_{i1}y_i \\
&\vdots \\
\hat{\beta}_0\sum_{i=1}^{n} x_{ik} + \hat{\beta}_1\sum_{j=1}^{n} x_{ik}x_{i1} + \cdots + \hat{\beta}_k\sum_{j=1}^{n} x_{ik}^2 &= \sum_{i=1}^{n} x_{ik}y_i
\end{aligned} \tag{2.7}$$

There are $p = k + 1$ least-squares normal equations, one for each of the $\hat{\beta}_j$ least-squares regression estimators. It is convenient to express the least-squares modeling equations as matrix equations. We can say that

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{2.8}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{2.9}$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \tag{2.10}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} \tag{2.11}$$

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \tag{2.12}$$

In general, $\mathbf{y}$ is an $n \times 1$ vector of observations of the dependent variable, $\mathbf{X}$ is an $n \times p$ matrix of observations of the regressor variables, $\boldsymbol{\beta}$ is a $p \times 1$ vector of the regression coefficients, and $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of random errors, assumed to be normally and independently distributed with mean $\mathbf{0}$ and constant variance. In terms of matrix notation, the least-squares procedure seeks to minimize the sum of squared error:

$$S(\boldsymbol{\beta}) = \sum_{i=1}^{n} \varepsilon_i^2 = \boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\mathbf{y} - \boldsymbol{\beta})'(\mathbf{y} - \boldsymbol{\beta}) \tag{2.13}$$

The least squares estimator of the regression coefficients $\boldsymbol{\beta}$ is given by.

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{2.14}$$

This equation is true as long as the $(\mathbf{X}'\mathbf{X})^{-1}$ exists. This will be the case when the regressor variables included in the model do not show strong linear dependencies.

The fitted linear regression model that arises from this least-squares analysis is

$$\hat{\mathbf{y}} = \mathbf{x}'\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \sum_{j=1}^{k} \hat{\beta}_j x_j \tag{2.15}$$

In terms of physical interpretation, the $\hat{\beta}_j$ values represent parameter coefficients that connect physical theory to experimental data. For example, in the molecular spectroscopy paper by Kurlat et al. (1971), the $\hat{\beta}_j$ are correspond to the molecular Hamiltonian parameter values retrieved from the analysis of observed spectral data.

2.3 Error Estimation

The vector of fitted values $\hat{y}_i$ that corresponds to the observed values y_i is

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{y} = \mathbf{H}\mathbf{y} \tag{2.16}$$

The $n \times n$ matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}$ is called the hat matrix, which maps the vector of fitted values into a vector of observed values.

The difference between an observed value y_i and the corresponding fitted value $\hat{y}_i$ is called the residual $e_i = y_i - \hat{y}_i$. The n residuals can be written as a column vector

$$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} \tag{2.17}$$

We note that there are several other ways of expressing the residuals, as shown below.

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{y} - \mathbf{H}\mathbf{y} = (\mathbf{I} - \mathbf{H})\mathbf{y} \tag{2.18}$$

We can now understand some statistical properties of the least-squares estimator $\hat{\boldsymbol{\beta}}$. Let us compute the expected value of $\hat{\boldsymbol{\beta}}$.

$$\begin{aligned} E(\hat{\boldsymbol{\beta}}) &= E\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}\right] \\ &= E\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon})\right] \\ &= E\left[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}\right] \\ &= \hat{\boldsymbol{\beta}} \end{aligned} \tag{2.19}$$

This result follows because $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = \mathbf{I}$ and $E(\boldsymbol{\varepsilon}) = \mathbf{0}$. Hence, we can see that $\hat{\boldsymbol{\beta}}$ is an unbiased estimator of $\boldsymbol{\beta}$. Furthermore, it can be proved via the Gauss-Markov theorem that $\hat{\boldsymbol{\beta}}$ is the best unbiased linear estimator of $\boldsymbol{\beta}$ by showing that $\hat{\boldsymbol{\beta}}$ has the smallest variance of all unbiased estimators that are linear combinations of the data. The relationships between the regression coefficients $\hat{\boldsymbol{\beta}}$ can be expressed as a covariance matrix

$$\begin{aligned} Cov(\hat{\boldsymbol{\beta}}) &= E\left[\left(\hat{\boldsymbol{\beta}} - E(\hat{\boldsymbol{\beta}})\right)\left(\hat{\boldsymbol{\beta}} - E(\hat{\boldsymbol{\beta}})\right)'\right] \\ &= \sigma^2(\mathbf{X}'\mathbf{X})^{-1} \end{aligned} \tag{2.20}$$

This is a $p \times p$ symmetric matrix where the diagonal elements are the variances of the respective estimators $\hat{\boldsymbol{\beta}}$ and the off-diagonal elements are the covariances between the respective estimators.

In order to obtain an estimate of σ^2, we can define a quantity call the *residual sum of squares*.

$$\begin{aligned} SS_{Res} &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} e_i^2 \\ &= \mathbf{e}'\mathbf{e} \end{aligned} \tag{2.21}$$

The residual sum of squares becomes

$$SS_{Res} = \mathbf{y}'\mathbf{y} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} \tag{2.22}$$

It can be shown that the residual sum of squares has $n-p$ degrees of freedom when we estimate p regression coefficients. This follows from the fact that the maximum number of regression coefficients (counting the intercept term) is equal to the number of data points. If the number of regression coefficients is equal to the number of data points, then this is the case where the fitted curve snakes through all of the data points and there is no estimation of error. The researcher generally tries to calculate fewer regression coefficients than there are data points, and the difference between these values defines the number of degrees of freedom for error.

We can define the *residual mean square* as

$$MS_{Res} = \frac{SS_{Res}}{n-p} \tag{2.23}$$

It can be shown that MS_{Res} is an unbiased estimator of σ^2 so that we have

$$\hat{\sigma}^2 = MS_{Res} \tag{2.24}$$

2.4 Regression Modeling Assumptions

After researchers calculate a model that they believe adequately describes the system under study, they must ensure that the modeling assumptions have been satisfied. When researchers use regression, they implicitly assume the following:

1. The errors in the regression model follow a normal distribution.

2. The errors have a constant variance σ^2.

3. The errors are uncorrelated.

These assumptions should be checked after every regression model is calculated, giving confidence to the conclusions of the analysis procedure.

In order to check if the residuals are normally distributed, we can plot the residuals on a *normal probability* plot. If we let $e_1 < e_2 < \cdots < e_n$ be the residuals ranked by increasing order, then we can plot e_i against the cumulative probability probability $P_i = (i - \frac{1}{2})/n$, $i = 1, 2, \ldots, n$. If the residuals are normally distributed, then this plot will be approximately a straight line. The analyst can visually inspect the normal probability plot to check deviations from normality. Gross departures from normality appear as "S" shapes in the plot and nonlinearities near the ends of the plotted line.

The constant variance assumption and autocorrelation assumption can be inspected by plotting the residuals e_i against the corresponding fitted values $\hat{y}_i$. In the ideal case, this plot should resemble points that are uniformly distributed around the zero centerline. No trends should appear in this plot. Problems with nonconstant

variance appear as funnel shapes, bows or arches, while autocorrelations show significantly more positive than negative values, or vise-versa. This type of plot can also identify outliers as isolated points that are unusually far from the centerline. If we transform the residuals as

$$r_i = \frac{e_i}{\sqrt{MS_{\text{Res}}(1 - h_{ii})}} \tag{2.25}$$

where h_{ii} is the ith diagonal element of the of the hat matrix $\mathbf{H}$, then these values are the *studentized* residuals, which have units of standard deviation. Problems with nonconstant variance often are more striking when plotted using studentized residuals. A value of r_i greater than approximatly 2.5 to 3 should be examined as an outlier.

Some problems with assumptions can often be corrected by performing a transformation. A logarithmic transformation or weighting scheme can correct for problems in the assumptions. Issues related to nonadherence to assumptions can be found in the literature.

2.5 Confidence Interval Estimation

In addition to estimating the values of the regression coefficients and associated variances, the researcher often needs to construct confidence intervals for the parameters. To this end, we continue to assume that the errors are normally and independently distributed. Let C_{jj} be the jth diagonal element of $(\mathbf{X}'\mathbf{X})^{-1}$. It can be shown via sampling theory that the value

$$\frac{\hat{\beta}_j - \beta_j}{\sqrt{\hat{\sigma}^2 C_{jj}}}, \ j = 0, 1, \ldots, k \tag{2.26}$$

has a t distribution with $n - p$ degrees of freedom. This allows us to construct a $100(1 - \alpha)$ confidence interval for the regression coefficient β_j.

$$\hat{\beta}_j - t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2 C_{jj}} \leq \beta_j \leq \hat{\beta}_j + t_{\alpha/2,n-p}\sqrt{\hat{\sigma}^2 C_{jj}} \tag{2.27}$$

Note that statisticians often call the quantity $\sqrt{\hat{\sigma}^2 C_{jj}}$ the *standard error* of the regression coefficient $\hat{\beta}_j$.

$$se(\hat{\beta}_j) = \sqrt{\hat{\sigma}^2 C_{jj}} \tag{2.28}$$

The procedure using the t distribution works well if the researcher is only interested in estimating confidence intervals for one regression coefficient at a time. Often, the researcher would like to estimate simultaneous confidence intervals for several regression coefficients. We can again appeal to sampling theory to construct such simultaneous confidence intervals. It can be shown that the $100(1 - \alpha)$ percent joint confidence region for all of the $\boldsymbol{\beta}$ parameters is:

$$\frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})' \mathbf{X}'\mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{p M S_{Res}} \leq F_{\alpha,p,n-p} \tag{2.29}$$

Geometrically, this inequality defines ellipsoids or hyperellipsoids centered on the regression coefficients $\boldsymbol{\beta}$. The ellipsoids are nested for decreasing values of the α parameter. This means the 99% confidence region encompasses more volume than the 95% confidence region, which itself surrounds the 90% confidence region, and so on. Most researchers regard an α value of 0.05 as a normal level of confidence in the process of statistical analysis. An α value of 0.01 is considered a strict threshold, while $\alpha = 0.10$ is a liberal value. Classically speaking, the α parameter defines the threshold of confidence that the researcher accepts or rejects hypotheses about the phenomenon under study.

We should remark that, while confidence intervals carry a frequentist interpretation, they are, strictly speaking, not probabilities. In a classical interpretation, if we repeat an experiment n times, the estimated parameter should fall within the calculated confidence interval $n(1-\alpha)$ times on average. However, the probability that the parameter is actually within this interval is either 0 or 1.

2.6 Sum of Squares

Let us define the *total sum of squares* as

$$\begin{aligned} SS_{\mathrm{T}} &= \sum_{i=1}^{n} y_i^2 - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n} \\ &= \mathbf{y}'\mathbf{y} - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n} \end{aligned} \tag{2.30}$$

We can also define the *regression sum of squares* as

$$SS_{\mathrm{R}} = \hat{\boldsymbol{\beta}}^{'}\mathbf{X}'\mathbf{y} - \frac{\left(\sum_{i=1}^{n} y_i\right)^2}{n} \tag{2.31}$$

It can be proved by a Pythagorean Theorem type argument that in a regression context, the total sum of squares can be partitioned into the regression sum of squares and the residual sum of squares.

$$SS_{\mathrm{T}} = SS_{\mathrm{R}} + SS_{\mathrm{Res}} \tag{2.32}$$

This equation provides some guidance about how to judge the quality of a model. The total sum of squares is conserved in a data set, but a model that accounts for more of the total in the regression sum of squares might be superior to another model

that has a smaller value for the regression sum of squares. We can also interpret it as follows; The total variance in the data set is constant, but the error variance can range from almost the entire variance to zero. The error variance can be visualized as the distribution of points surrounding the fitted line. A small error variance appears as a cigar shaped distribution hugging the fitted line while a larger error variance becomes a cloud of points drifting far away from it.

The question can be raised about how to judge competing models which may include different combinations of variables. We could use the relative sizes of SS_{R} and SS_{Res} as a guide rate the quality of the model. One criterion is R^2, which is the ratio of the regression sum of squares to the total sum of squares. Another criterion is adjusted R^2, which is corrected for error degrees of freedom. These values can be summarized as:

$$\begin{aligned} R^2 &= \frac{SS_{\mathrm{R}}}{SS_{\mathrm{T}}} \qquad (2.33) \\ &= 1 - \frac{SS_{\mathrm{Res}}}{SS_{\mathrm{T}}} \\ R^2_{adj} &= 1 - \frac{SS_{\mathrm{Res}}/(n-p)}{SS_{\mathrm{T}}/(n-1)} \qquad (2.34) \end{aligned}$$

We note that R^2 can assume values from 0 to 1. This might be regarded as a measure of the quality of the regression equation, with higher values of R^2 denoting better models.

2.7 Significance of Regression

After the regression coefficients have been estimated, the researcher would like to test if there is a linear relationship between the response variable y and any regressor variables $x_1, x_2, \ldots, x_k$. Many statisticians call this testing the "significance of the

regression." In this sense, the relationship is significant if the statistical analysis shows that it does not occur by chance alone. In a classical statistical framework, testing significance means calculating a hypothesis test, where the null hypothesis is denoted H_0 and the alternative hypothesis is denoted H_1. In order to test the overall significance of a model, we can state the hypothesis as:

$$\begin{aligned} H_0 &: \beta_1 = \beta_2 = \cdots = \beta_k = 0 \\ H_1 &: \beta_j \neq 0 \text{ for at least one } j \end{aligned}$$

We can interpret this test in terms of a logical argument. The null hypothesis states that all β values are equal to 0. The logical negation of this statement is that at least one β value is nonzero. Rejecting the null hypothesis and accepting the alternative hypothesis means that at least one of the regressor variables contributes to a linear relationship with the response variable. Classical statistics has a procedure called Analysis of Variance or ANOVA, that tests the overall significance of the regression model.

Continuing the argument of testing significance of an overall model, if the null hypothesis is correct, then it can be shown that the test statistic

$$\begin{aligned} F_0 &= \frac{SS_{\text{R}}/k}{SS_{\text{Res}}/(n-k-1)} \\ &= \frac{MS_{\text{R}}}{MS_{\text{Res}}} \\ &\sim F_{k,n-k-1} \end{aligned} \tag{2.35}$$

where MS_{R} and MS_{Res} are the *regression mean square* and the *residual mean square* respectively. Hence, we can use this value to perform an overall test of significance. If $F_0 > F_{\alpha,k,n-k-1}$, then we reject the null hypothesis and conclude that at least

one of the regressor variables contributes significantly to a linear relationship with the response y. The $F_{\alpha,k,n-k-1}$ values can be found in common tables of the F distribution for given values of α.

2.8 Variable Selection

In addition to testing the overall significance of the model, the researcher would like to know which regressor variables are related to the response. We can recall that one of the main goals of statistical modeling is to decide which variables are most important in describing a phenomenon under study. As a first attempt at variable selection, classical statistical methodology advocates hypothesis testing on individual coefficients. The researcher can perform hypothesis tests on individual regression coefficients such that

$$\begin{aligned} H_0 &: \beta_j = 0 \\ H_1 &: \beta_j \neq 0 \end{aligned}$$

If the null hypothesis is rejected, then we can add the regressor x_j to the model. To accomplish this hypothesis test, we can compute the test statistic

$$t_0 = \frac{\hat{\beta}_j}{\sqrt{\hat{\sigma}^2 C_{jj}}} = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)} \tag{2.36}$$

and reject the null hypothesis $H_0 : \beta_k = 0$ if $|t_0| > t_{\alpha/2,n-k-1}$.

While this appears to be a simple procedure for computing which variables to include in the model, there are many subtleties implicit in this process. The multivariate least-squares equations were derived assuming that the columns in the $(\mathbf{X}'\mathbf{X})^{-1}$ were linearly independent. The researcher must realize that, unless the data comes from a

designed experiment with orthogonal variables, the data will exhibit some linear dependence. However, the degree to which the variables are linearly dependent greatly affects the conclusions reached by least-squares analysis. In statistics, this linear dependence is called multicollinearity. Including variables with multicollinearity leads to problems that include inflated variance estimates of the regressor variables and instabilities in hypothesis test values. In many physical and astronomical data analysis situations, the observed data does not come from an orthogonal experiment. Therefore, the t-test procedure is actually a partial t test because, in general, the regression coefficient $\hat{\beta}_j$ depends on which variables are already included in the model. This is a test of the amount variable x_j contributes to the model given the other regressors in the model.

Another obstacle for researchers is how to choose the best model as the number of possible variables grows. One way to find the best model is to produce all combinations of the possible variables for a regression model, and rank the models by some criterion, like R^2. We can observe that for k regressor variables, the number of possible models grows like 2^k, so combinatorial all-regressions analysis becomes impractical even for modest values of k.

Another method of searching for a good model is to use stepwise regression. The algorithm that is called "stepwise regression" was introduced by Efroymson (1960). This stepwise regression, or slight modifications of it, have historically been the method of choice among researchers modeling highly multivariate phenomena (Boyd 1963, Blass 1963, M. Kurlat 1969, H. Kurlat 1970, Hafford 1972). Before the stepwise regression loop begins, the user defines F_{in} and F_{out}. Efroymson states that F_{out} should be greater than F_{in}. At each iteration, the algorithm computes an F value for variables to enter and an F value for variables to leave. For residual sum of squares

SS_{Res} and a model with p parameters, the F to enter value is

$$F = \frac{SS_{\text{Res,p}} - SS_{\text{Res,p+1}}}{SS_{\text{Res,p+1}}/(n-p-1)} \tag{2.37}$$

Likewise, the F to leave value is

$$F = \frac{SS_{\text{Res,p-1}} - SS_{\text{Res,p}}}{SS_{\text{Res,p}}/(n-p)} \tag{2.38}$$

Miller (1996) reconsidered Efroymson's stepwise algorithm, critically examining its convergence properties. Miller summarizes the stepwise regression algorithm as follows:

1. Insert variables that the researcher believes are important. These variables are forced to be in the model at each iteration of the algorithm.

2. Find the variable that is not currently in the model that has the largest F statistic value to enter. If there are no variables in the model that have an F value as large as the F_{in} value, then stop.

3. Find the variable in the model, other than those forced to be in, that has the smallest F statistic value to remove. If this value is less than F_{out}, then remove this variable from the model. Repeat this procedure until no more variables are dropped, then go to step 2.

The advantage of this stepwise procedure is that it can efficiently process a large number of possible models. The process where no variables are initially in the model is called *forward stepwise regression*, while the process where all variables are initially in the model is called *backwards stepwise regression*.

2.9 Regression Through the Origin

Some modeling situations do not have an intercept term. Hahn (1977) describes several situations where the interpretation of the model does not require adding an intercept term. Neter et al. (1990) defines the regression through the origin model as

$$y_i = \beta x_i + \varepsilon_i \tag{2.39}$$

The regression function in this case is

$$E(y) = \beta x \tag{2.40}$$

We then minimize the Sum of Squared Error to be

$$SS_{Res} = \sum_{i=1}^{n} (y_i - \beta x_i)^2 \tag{2.41}$$

which leads to the least squares normal equation

$$\sum_{i=1}^{n} x_i (y_i - \beta x_i) = 0 \tag{2.42}$$

with regression contant estimate

$$\beta = \frac{\sum_i x_i y_i}{\sum_i x_i^2} \tag{2.43}$$

The mean squared error estimate of σ^2 then becomes

$$\begin{aligned} \hat{\sigma}^2 &= MS_{Res} \\ &= \frac{\sum_{i=1}^{n} (y_i - \beta x_i)^2}{n-1} \end{aligned} \tag{2.44}$$

The development of regression without an intercept easily generalizes to the multivariate case.

2.10 Relation to Chi-squared Analysis

The method of linear regression can also be derived from the perspective of chi-squared analysis. In general, the chi-squared test is used to compare theoretical predictions to observed data points. A commonly used test to compare observed data to expected data is Pearson's chi-squared statistic. If the physical model has observed count data O and expected count data E, then the goodness-of-fit test (Rice 1995) is for n data points is

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)}{E_i} \tag{2.45}$$

Let a set of measurements y_i have variances σ_i^2. In terms of comparing departures of data points from their mean value μ, we have that

$$\chi^2(\mu) = \sum_{i=1}^{n} \frac{(y_i - \mu)^2}{\sigma_i^2} \tag{2.46}$$

A change in the the fit of the data by one standard deviation corresponds to a change in the chi-squared statistic of 1, as we can show.

$$\begin{aligned} \chi^2(\mu + \sigma_\mu) &= \sum_{i=1}^{n} \frac{(y_i - (\mu \pm \sigma_\mu))^2}{\sigma_i^2} \\ &= \sum_{i=1}^{n} \frac{(y_i - \mu)^2 \pm 2\sigma_\mu (y_i - \mu) + \sigma_\mu^2}{\sigma_i^2} \\ &= \chi^2(\mu) + \sigma_\mu \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \end{aligned} \tag{2.47}$$

By the definition of variance, we also have

$$\sigma_\mu = \frac{1}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}} \tag{2.48}$$

Hence, we have the result

$$\chi^2(\mu + \sigma_\mu) = \chi^2(\mu) + 1 \tag{2.49}$$

Turning to the case of simple regression, let $f(x)$ fit dependent measurements $\{y_i\}$ with independent measurements $\{x_i\}$. For n measurements, each with standard deviation σ, then the variance of the best-fit relationship is

$$\begin{aligned} V &= \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x)) \\ &= \frac{1}{n}\sigma^2\chi^2 \end{aligned} \tag{2.50}$$

For multiple linear regression, if there are p parameters to estimate, then the expected value of the variance of errors is

$$\langle V \rangle = \sigma^2 \frac{n}{n-p} \tag{2.51}$$

It can be shown that this relationship follows a chi-squared distribution with $n-p$ degrees of freedom.

These properties allow us to relate the χ^2 distribution to the F distribution. The sample standard deviation of the fit with ν degrees of freedom is

$$\begin{aligned} s^2 &= \frac{1}{\nu}\sum_{i=1}^{n}(y_i - f(x_i))^2 \\ &= \frac{\sigma^2}{\nu}\chi^2 \end{aligned} \tag{2.52}$$

For two samples from the same population that have variance σ^2, the F distribution is

$$\begin{aligned} F &= \frac{s_1^2}{s_2^2} \\ &= \frac{\chi_1^2/\nu_1}{\chi_2^2/\nu_2} \end{aligned} \tag{2.53}$$

In multiple linear regression, we can define the function $f(x)$ with parameters $\{\beta_k\}$.

$$f(x_i) = \bar{y} + \sum_k \beta_k (X_{ki} - \bar{X}_k) \tag{2.54}$$

here, X_{ki} is the kth function of x evaluated at x_i. We can define the χ^2 for this function to be

$$\chi^2 = \sum_{i=1}^{n} \frac{\left(y_i - \bar{y} - \sum_k \beta_k (X_{ki} - \bar{X}_k)\right)^2}{\sigma_i^2} \tag{2.55}$$

which has a least-squares solution of

$$\frac{\partial \chi^2}{\partial \beta_k} = -2\sum_i \frac{\left(y_i - \bar{y} - \sum_k \beta_k (X_{ki} - \bar{X}_k)\right)(X_{ki} - \bar{X}_k)}{\sigma_i^2} \tag{2.56}$$

we also have

$$\sum_i \frac{(y_i - \bar{y})\,(X_{ki} - \bar{X}_k)}{\sigma_i^2} = \sum_{i,k} \frac{\beta_k (X_{ki} - \bar{X}_k)(X_{ki} - \bar{X}_k)}{\sigma_i^2} \tag{2.57}$$

The chi-squared function at the best-fit solution then is

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \bar{y})^2}{\sigma_i^2} - \sum_{i=1}^{n} \frac{(y_i - \bar{y}) \sum_k \beta_k (X_{ki} - \bar{X}_k)}{\sigma_i^2} \tag{2.58}$$

Under the chi-squared derivation, we can define R^2 to be

$$R^2 = \frac{\sum_k \beta_k \sum_i \frac{(y_i - \bar{y})\beta_k (X_{ki} - \bar{X}_k)}{\sigma_i^2}}{\sum_i \frac{(y_i - \bar{y})^2}{\sigma_i^2}} \tag{2.59}$$

The chi-squared test statistic then becomes

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \bar{y})^2}{\sigma_i^2} (1 - R^2) \tag{2.60}$$

In linear regression, the error variance is constant, so the chi-squared becomes

$$\chi^2 = \frac{1}{\sigma_i^2} \sum_{i=1}^{n} (y_i - \bar{y})^2 \,(1 - R^2) \tag{2.61}$$

2.11 Conclusion

This chapter summarized methods of classical statistical regression, which rely on hypothesis tests and levels of significance to compute model equations. This chapter did not address some issues encountered in regression modeling, including how to deal with multicollinearity, calculating P-values, and optimizing experimental design. The interested reader can consult many texts and other published works on these topics.

Chapter 3

Linear Regression with Information Scores

3.1 Introduction

One of the main goals of scientific study is to analyze data and determine which measurable quantities are related to other measurable quantities. Statistical methods must be used to analyze these relationships. Linear regression is one tool that studies relationships between variables. If the researcher has at k regressor variables and one response variable, linear regression calculates parameters of an equation with the form

$$y = \beta_0 + \beta_1 x_1 + \beta_1 x_1 + \ldots + \beta_k x_k \tag{3.1}$$

Classical linear regression has historically been a favorite method among researchers. However, some subtleties are implicit in the classical regression method. In order to illustrate some of these inherent assumptions, consider a published example where Jefferys and Berger (1992) described the situation of Galileo analyzing data from a

falling body. Galileo introduced the equation describing the distance s that a body falls during a specified time t as

$$s = a + ut + \frac{1}{2}gt^2 \tag{3.2}$$

where constants a, u and g can be assigned empirically from data. This expression describes the observed relationship well and is able to predict new observations from future experiments. However, from the perspective of data analysis, if the goal of the model is to reduce the sum of squared error between the observed data and the statistical model, there is no reason to stop with this equation. An equation of the form

$$s = a + ut + \frac{1}{2}gt^2 + bt^3 \tag{3.3}$$

has a smaller sum of squared error than the previous equation. In fact, the classical statistical test of R^2 regards this as a better description of the data than the second-order law. Fourth or sixth order polynomials account for even more error than the third-order polynomial. This process begs the question: Why do physicists prefer the second-order law?

William of Ockham might answer this question with "Pluralitas non est ponenda sine necessitate", which translates as "Plurality must not be posited without necessity" (Jefferys and Berger 1992). Ockham's razor is the principle that, given competing models that describe a process under study, the simplest one that adequately explains the phenomenon is the best. Some scientists might consider this to be a philosophical statement rather than a scientific principle. Ockham's razor, however, leads to the very practical and important Principle of Parsimony in scientific data analysis. The Principle of Parsimony restates Ockham's razor as a way of selecting

competing statistical models of the same data set. The least complex statistical model is the best one that describes the system under study.

The example of Galileo's falling body illustrates a dilemma of using classical regression methods. Classical regression is based on reducing the sum of squared error in a statistical model. This strategy works well for univariate modeling situations. The goal of univariate regression modeling is to calculate a slope and intercept that minimizes error variance. However, in a multivariate modeling situation, reducing the sum of squared error can lead to ambiguity in model selection. Researchers may use R^2 as a basis for comparing model quality. However, adding more variables to a regression model only increases R^2. The value of R^2 will generally sharply rise as the first few variables are added to the model, and then flattens into a gently-sloping plateau. Researchers using classical regression might try to find the point where R^2 plateaus. This analysis is a subjective exercise, where different researchers will disagree about the best model. In addition, in cases with a large number of variables, many combinations of variables can make R^2 exhibit this kind of behavior, and the analyst must rely on their own judgement to find the best model.

Another issue confronting classical regression is overfitting. Overfitting involves the inclusion of redundant variables, which leads to models that do not generalize well and cannot adequately describe new data points from the same process. Overfitting goes against the Principle of Parsimony by including unnecessary variables. The methods used by classical regression seek to minimize sum of squared error, which generally overfit the computed models.

The problem of selecting an optimal model becomes a bigger challenge as the number of variables increases. For decades, researchers have called this the "Curse of Dimensionality" (Bellman 1961). The total number of models for k possible regression

variables is $2^k - 1$. This large number makes evaluating all combinations impractical even for relatively small values of k.

Historically, researchers used stepwise regression to process a large number of variables. Efroymson's algorithm, introduced in 1960, was one of the first widely used stepwise regression procedures. Efroymson's (1960) procedure was a forward stepwise procedure. However, within 10 years of its publication, some major criticisms appeared in statistical literature. Mantel (1970) evaluates the merits of backward regression as opposed to forward regression. It is easy to demonstrate that forward stepwise regression and backward stepwise regression generally give different results even though they operate on the same data set. Other authors (Boyce et al. 1974, Wilkinson 1989, pg 177-178) criticize stepwise regression because of the arbitrary thresholds of variables to enter or leave the model. Efroymson suggests a value of 4.0 for both F_{in} and F_{out}, but does not comment on why this value has merit or how it affects the termination criterion (Miller 1996). Blass (private communication) uses a value of approximately 3 in his stepwise algorithm. Changing the entry or exit threshold values generally changes the final model computed by the stepwise procedure. Boyce and Wilkinson continue by writing that there is little to no theoretical justification for using any form of stepwise regression. Mantel (1970), Hocking (1976) and Moses (1986) level further criticisms against stepwise regression by saying that it rarely finds the optimal model even when restricted to a subset of variables. Perhaps the most serious shortcoming of stepwise regression is its localized searching method. Sokal and Rolf (1981, pg 668) write that its limited searching area generally produces only "adequate" models. The compounded effect of all these issues leads to doubt about how reliably researchers can draw physical interpretations from stepwise analysis. Some researchers may justify why they set the thresholds as some level or

have some predefined order for variable selection, but nothing fundamental about the stepwise process justifies these choices.

To address these inherent flaws in classical statistical regression, Akaike (1973) introduced a penalized scoring function to evaluate models. Called Akaike Information Criteria (AIC), it is expressed as

$$AIC = -2logL(\Theta_k) + 2m(k) \tag{3.4}$$

where $logL(\Theta_k)$ is the maximized log likelihood of the regression and $m(k)$ is the number of free parameters in the model. These two terms have opposite signs and act like opposing forces during the modeling process. AIC assigns scores to different combinations of parameters. The combination that achieves the best balance between accounting for error and including enough terms in the model has the lowest score.

Since Akaike's landmark paper, researchers have introduced other penalized measures to calculate parsimonious regression models. Some of these include Rissanen's (1978, 1986) Minimum Description Length (MDL), Schwartz's (1978) Bayesian Information Criterion (BIC), Bozdogan's (Consistent AIC with Fisher information (CAICF) (1987) and Bozdogan's (1988, 1990a, 2000, 2004) ICOMP. All of these calculate scores for different combinations of model parameters, trying to include enough model parameters to accurately describe the systems under study while guarding against overfitting.

ICOMP extends AIC from simply subtracting the number of included terms to analyzing interactions of the included components. Quantifying the interactions of the included model terms gives a measure of the complexity of the system. ICOMP measures the structural complexity of the system by estimating a loss function of the form

Loss = Lack of Fit + Lack of Parsimony + Profusion of Complexity

This expression generalizes Van Emden's (1971) information based covariance complexity index. Using this definition, we can derive a numerical score that quantifies the amount of complexity in a statistical model. Bozdogan (1988) proposed ICOMP to be

$$ICOMP = -2logL(\Theta_k) + 2C_1(\Sigma_{Model}) \tag{3.5}$$

The first component is the log-likelihood of the model, identical to the first term in AIC. The second component, however, measures the covariance complexity of a model by calculating a scaler index for the covariance matrix Σ_{Model}. These two terms counteract each other when scoring model combinations. The model with the minimum ICOMP score is the most parsimonious that adequately describes the system under study. In addition, because AIC penalizes only the number of included components, but does not consider their interactions, it has been shown that AIC can overfit models. ICOMP overcomes the overfitting problem of AIC, and is considered by experts to be the most modern and accurate method of statistical modeling (Bozdogan, private communication).

The remainder of this chapter derives ICOMP as a regression model scoring function. We start by reviewing the method of maximum likelihood estimation which is used in the first term of both AIC and ICOMP. Later, we develop the concept of complexity from a set-theoretic perspective, and then apply this to Fisher information scoring as an index of covariance complexity.

3.2 Maximum Likelihood Estimation

Maximum likelihood plays a central role in information based statistics. In evaluation functions like AIC and ICOMP, maximum likelihood acts as the Lack of Fit component of the equation. Modeling based on maximum likelihood estimation can

be more general than least-squares methods because it can incorporate many different error distributions, and it can be implemented in complex modeling situations. Suppose that random variables $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ have joint density function $f(x;\theta)$, where x is an observed data point and θ is a parameter that should be estimated. Given the data set $\mathbf{X} = (x_1, x_2, \ldots, x_n)$, we can define the likelihood function as

$$L(\theta|\mathbf{x}) = \prod_{i=1}^{n} f(x_i;\theta) \tag{3.6}$$

Note that the likelihood function is a real-valued function for every possible data sample $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

It is generally more convenient to work with the natural logarithm of the likelihood instead of the likelihood itself. Because log is a monotonic function, the log-likelihood can also be used to estimate parameters. The log-likelihood is defined as

$$l(\theta) = \sum_{i=1}^{n} \ln f(x_i;\theta) \tag{3.7}$$

Both the likelihood and log-likelihood measure the plausibility of estimating unknown parameters of a data set. Likelihood and probability are closely related, but have different interpretations. Probability estimates what data will be produced given a set of parameters, while likelihood estimates what parameters will arise from a given data set.

Once the likelihood or log-likelihood function has been defined, we can turn to *maximum likelihood estimation*. The maximum likelihood estimate gives the value of the parameter that has the highest probability of yielding the observed data set. The maximum likelihood estimate of parameter $\hat{\theta}$ can be derived as the stationary value of θ such that

$$\frac{dl(\theta)}{d\theta} = 0 \tag{3.8}$$

We can illustrate maximum likelihood estimation with a few examples. Consider data observed from an exponential distribution where $X_1, X_2, \ldots X_n \sim Exp(\lambda = \theta)$. In this case, the observations X_i follow the distribution

$$f(x_i \mid \theta) = \theta e^{-\theta x_i} \ , \ (x_i > 0) \tag{3.9}$$

We can derive the likelihood as

$$L(\theta) \ = \ \prod_{i=1}^{n} \theta e^{-\theta x_i} \tag{3.10}$$

and the log-likelihood

$$\begin{aligned} l(\theta) &= \ln L(\theta) \\ &= -n \ln \theta + \theta \sum_{i=1}^{n} x_i \end{aligned} \tag{3.11}$$

the stationary point becomes

$$\frac{dl(\theta)}{d\theta} = \frac{n}{\theta} - \sum_{i=1}^{n} x_i = 0 \tag{3.12}$$

$$\hat{\theta} = \frac{n}{\sum_{i=1}^{n} x_i} = \frac{1}{\bar{x}} \tag{3.13}$$

where $\bar{x}$ is the mean of the data set

Next, we can examine data that arise from a Poisson distribution. Suppose that observations $x_1, x_{2,\ldots,}x_n$ come from the Poisson distribution $X_1 \ , X_2, \ldots, X_n \sim P(\lambda = \theta)$.

$$f(x_i \mid \theta) = \frac{\theta^{x_i} e^{-\theta}}{x!}, \ x = 0, 1, 2, \ldots \tag{3.14}$$

Then the analysis is as follows. We may disregard any constant multiplier that does not depend on θ

$$L(\theta) = \prod_{i=1}^{n} \theta^{x_i} e^{-\theta}, \ (x_i > 0) \tag{3.15}$$

$$\begin{aligned} l(\theta) &= \ln L(\theta) \\ &= -n\theta + \sum_{i=1}^{n} x_i \ln(\theta) \end{aligned} \tag{3.16}$$

$$\frac{dl(\theta)}{d\theta} = -n + \frac{\sum_{i=1}^{n} x_i}{\theta} = 0 \tag{3.17}$$

which yields the Maximum Likelihood Estimate (MLE) of

$$\hat{\theta} = \frac{\sum_{i=1}^{n} x_i}{n} = \bar{x} \tag{3.18}$$

In most regression modeling situations, we consider data that follow a normal distribution. Suppose that a data set $x_1, x_2, \ldots, x_n \sim N(\theta, 1)$, and we would like to estimate the mean θ. We can derive the MLE of normally distributed data in a similar way.

$$f(x_i \mid \theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_i - \theta)^2\right) \tag{3.19}$$

Then

$$L(\theta) \propto \prod_{i=1}^{n} \exp\left(-\frac{1}{2}(x_i - \theta)^2\right) \tag{3.20}$$

$$\begin{aligned} l(\theta) &= \ln L(\theta) \\ &= -\frac{1}{2} \sum_{i=1}^{n} (x_i - \theta)^2 \end{aligned} \tag{3.21}$$

$$\frac{dl(\theta)}{d\theta} = \sum_{i=1}^{n} (x_i - \theta) = 0 \tag{3.22}$$

So

$$\hat{\theta} = \frac{\sum_{i=1}^{n} x_i}{n} = \bar{x} \tag{3.23}$$

Maximum likelihood estimation can be applied to regression modeling and to estimating multiple parameters simultaneously. If we consider a simple linear regression where we have data points (y_i, x_i), $i = 1, 2, \ldots, n$, and we assume that the errors are normally and independently distributed (NID), and we would like to estimate the slope β_0, the intercept β_1 and the variance σ^2. The likelihood function then is

$$\begin{aligned} L(y_i, x_i|\beta_0, \beta_1, \sigma^2) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2\right) \end{aligned} \tag{3.24}$$

We can write the log-likelihood function that estimates parameter values $\hat{\beta}_0, \hat{\beta}_1$, and $\hat{\sigma}^2$.

$$\begin{aligned} \ln L(y_i, x_i|\beta_0, \beta_1, \sigma^2) &= -\frac{n}{2}\ln 2\pi - \frac{n}{2}\ln\sigma^2 \\ &\quad -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2 \end{aligned} \tag{3.25}$$

The MLE must satisfy the simultaneous stationary points such that

$$\begin{aligned} \frac{\partial \ln L}{\partial \beta_0}_{\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2} &= \frac{1}{\hat{\sigma}^2}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right) = 0 \\ \frac{\partial \ln L}{\partial \beta_1}_{\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2} &= \frac{1}{\hat{\sigma}^2}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right)x_i = 0 \\ \frac{\partial \ln L}{\partial \sigma^2}_{\hat{\beta}_0, \hat{\beta}_1, \hat{\sigma}^2} &= -\frac{n}{2\hat{\sigma}^2} + \frac{1}{\hat{\sigma}^4}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i\right)^2 = 0 \end{aligned} \tag{3.26}$$

We can solve these equations to yield the MLE for the parameters:

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x} \\ \hat{\beta}_1 &= \frac{\sum_{i=1}^{n} y_i (x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \\ \hat{\sigma}^2 &= \frac{\sum_{i=1}^{n} (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2}{n} \end{aligned} \tag{3.27}$$

The generalization of MLE for multivariate regression modeling is straightforward and proceeds in the same way.

3.3 Fisher Information in Linear Regression

In order to understand the statistical properties of MLE's, we can define the Fisher information. Fisher information studies the variance of terms in a model, and is related to the amount of information contained in a data set. Consider variables $X = (X_1, X_2, \ldots, X_n)$ with joint probability function $f(x \mid \theta)$. We can then define a score function as

$$S(\theta) = \frac{df(x \mid \theta)}{d\theta} = \frac{dl(\theta)}{d\theta} \tag{3.28}$$

This score function is the gradient of the log-likelihood function. By the definition of MLE, we know that $S(\hat{\theta}) = 0$. We can think of θ as the true value of the parameter. $S(\theta)$ will be negative if $\hat{\theta}$ underestimates θ and $S(\theta)$ will be positive if $\hat{\theta}$ overestimates θ. It can be shown that

$$E\left[S(\theta)\right] = 0 \tag{3.29}$$

$$Var\left(S(\theta)\right) = E\left[-\frac{dl(\theta)}{d\theta}\right] \tag{3.30}$$

The quantity $\mathbf{F}(\theta) = E\left[-\frac{dl(\theta)}{d\theta}\right]$ is called the *expected information* or the *Fisher information*, and it measures the expected curvature of the log-likelihood function at the true parameter value. We can prove that MLE's have minimum variance among all unbiased estimators by appealing to the Cramer-Rao theorem, which states that if $\hat{\theta}$ is an estimator of θ, then $Var(\hat{\theta}) \geq \mathbf{F}(\theta)^{-1}$. We can illustrate this property with an example. Suppose that we have a data set that comes from an exponential distribution. The log-likelihood function is

$$\begin{aligned} l(\theta) &= \ln L(\theta) \\ &= n \ln \theta - \theta \sum_{i=1}^{n} x_i \end{aligned} \tag{3.31}$$

The derivative then is

$$\frac{dl(\theta)}{d\theta} = \frac{n}{\theta} - \sum_{i=1}^{n} x_i \tag{3.32}$$

and

$$\frac{d^2 l(\theta)}{d\theta^2} = -\frac{n}{\theta^2} \tag{3.33}$$

so that

$$\begin{aligned} \mathbf{F}(\theta) &= E\left[-\frac{dl(\theta)}{d\theta}\right] \\ &= E\left[\frac{n}{\theta^2}\right] \\ &= \frac{n}{\theta^2} \end{aligned} \tag{3.34}$$

Then the Cramer-Rao bound for θ is

$$CRB(\theta) = \mathbf{F}^{-1}(\theta) = \frac{\theta^2}{n} \tag{3.35}$$

The sample estimate is given by

$$Esimated\ IFIM = \hat{\mathbf{F}}^{-1}(\hat{\theta}) = \frac{\hat{\theta}^2}{n} \tag{3.36}$$

It can be shown that as $n \longrightarrow \infty$

$$\hat{\theta} \sim N(\theta, \mathbf{F}^{-1}(\theta)) \tag{3.37}$$

As a result, we can show that the MLE is an unbiased, normally distributed estimator. It is also fully efficient in the sense that it reaches the Cramer-Rao lower bound. Sometimes, it is more convenient to use the *observed information* defined as

$$\mathbf{F}_{obs}(\theta) = \left[-\frac{dl(\theta)}{d\theta}\right]_{\theta=\hat{\theta}} \tag{3.38}$$

which is the observed curvature evaluated at the MLE. This lets us calculate confidence intervals for our regression.

$$\frac{\hat{\theta} - \theta}{\sqrt{\frac{1}{\mathbf{F}_{obs}(\theta)}}} \sim N(0, 1) \tag{3.39}$$

To calculate the approximate 95% confidence interval for θ, we can use the observed Fisher information. The value 1.96 comes from the value of a standardized normal distribution evaluated at the 95% confidence limits.

$$\theta \pm 1.96\sqrt{\mathbf{F}^{-1}(\theta)} \tag{3.40}$$

3.4 Information Scores in Regression Modeling

Using the machinery of maximum likelihood estimation, we can now analyze model selection by information criteria. Akaike (1973) introduced a penalized score function to evaluate the quality of proposed models. This function has a term that rewards the model for including terms that account for an observed relationship, but punishes the model for including extraneous terms. Akaike Information Criteria (AIC) can be written as

$$AIC = -2 \log L(\hat{\theta}) + 2k \tag{3.41}$$

In the AIC function, the $\log L(\hat{\theta})$ term is the maximized log-likelihood that acts as a lack of fit term, while k is the number of free parameters in the model, and acts as the penalty that guards against overfitting. AIC seeks a compromise between these competing terms. The model that has the minimum AIC score is the most parsimonious in describing the phenomenon under study.

As an example, we can list the AIC expressions for common distributions. The number of unknown parameters come from the respective distributions. The normal distribution, for example, has two unknown parameters (mean and variance) while the exponential distribution has only one unknown parameter λ. To calculate the value of AIC, the researcher would estimate the terms (like $\hat{\sigma}^2$ and $\bar{x}$) from their data set and substitute these terms into the expression for AIC.

$$\begin{aligned} AIC(Normal) &= n \ln(2\pi) + n \ln(\hat{\sigma}^2) + n + 2(2) && (3.42) \\ AIC(Exponential) &= 2n + 2n \ln(\bar{x}) + n + 2 \sum_{i=1}^{n} \ln(x_i) + 2(1) && (3.43) \end{aligned}$$

While AIC was a departure in the methodology used in statistical modeling, it does not always perform optimally in some modeling situations. It has been shown that

AIC tends to overfit data by including unnecessary terms. Bozdogan formulated the Consistent AIC (abbreviated CAIC) in 1987. In this work, the penalty term was modified to depend on the sample size n. CAIC can be defined as:

$$CAIC = -2\log L(\hat{\theta}) + k(\log n + 1) \tag{3.44}$$

ICOMP was developed to deal with complex multivariate modeling situations. ICOMP is written as

$$ICOMP = -2logL(\Theta_k) + 2C(\Sigma_{Model}) \tag{3.45}$$

where the maximized log-likelihood term performs the same role as in AIC. The complexity term computes a scaler measure of the interactions between the components in the model. The remaining parts of the chapter develop the notion of complexity and show the full derivation of the complexity operation for normal regression modeling. Pragmatically, in order to use ICOMP in a regression context, the researcher can simply implement the final derived expressions for ICOMP(REG) and ICOMP(IFIM):

$$ICOMP(REG) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2C_1((\mathbf{X}'\mathbf{X})^{-1}) \tag{3.46}$$

with

$$C_1((\mathbf{X}'\mathbf{X})^{-1}) = \frac{q}{2}\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1})}{q}\right] - \frac{1}{2}\log\left[\det((\mathbf{X}'\mathbf{X})^{-1})\right] \tag{3.47}$$

Here, $q = rank((\mathbf{X}'\mathbf{X})^{-1})$.

$$\begin{aligned} ICOMP(IFIM) &= n\log(2\pi) + (n - \frac{1}{2})\log(\hat{\sigma}^2) + n \\ &\quad +(q+1)\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1}) + \frac{2\hat{\sigma}^2}{n}}{q+1}\right] \\ &\quad -\log\left(\det\left((\mathbf{X}'\mathbf{X})^{-1}\right)\right) - \log\left(\frac{n}{2}\right) \end{aligned} \tag{3.48}$$

The regression algorithm assigns ICOMP scores to different combinations of parameters. The model that achieves the lowest score is the best to describe the system.

3.5 Measuring Complexity in Statistics

Information based statistics starts with an understanding of the concept of complexity. A general definition of complexity can be written as (Van Emden 1971): "Complexity of a system (of any type) is a measure of the degree of interdependency between the whole system and a simple enumerative composition of its subsystems or parts." This means that if we can decompose a system into subsystems and assign a score to each of the subsystems, we can gain a better understanding of how each part is related to the entire system. In this vein, we can recall a property of decomposition of sets. For a set that contains k members, there are 2^k possible subsets (including the empty set). If we apply this idea to linear statistical modeling, we can perform a power set decomposition on a set of variables, and assign a score to each possible subset to measure the complexity of the system. However, generally, we do not include the empty set in linear regression, so there are $2^k - 1$ variable combinations for k possible variables.

In order to calculate an information complexity score for a statistical system, we can appeal to information theory. Information theory arose from the study of

digital communication, where researchers tried to understand how information can be represented and reconstructed from digital signals. Information theory seeks to optimize the amount of information that can be transmitted in the fewest bits of signal. We can start with the definition of entropy of information. For probability distribution $p(x)$, a measure of the uncertainty of the probability distribution is the entropy, which is defined as

$$H(X) = -\sum_{x \in \Psi} p(x) \log(p(x)) = E\left[\log\left(\frac{1}{p(x)}\right)\right] \tag{3.49}$$

We note that since $0 < p(x) < 1$, then the $\log(p(x))$ is negative, so the entropy is positive. Entropy provides a measure of the sharpness of the probability distribution $p(x)$, which gives a notion of the uncertainty involved. If the entropy $H(X) = 0$, then this corresponds to a deterministic process with only one outcome. By contrast, the maximum entropy is achieved by a uniform distribution. It can be shown that a uniform distribution contains the maximum amount of uncertainty about a random variable.

If we have a pair of random variables X and Y, defined over domains Ψ and Φ respectively, then the joint probability $p(x, y)$ gives rise to joint entropy $H(X, Y)$.

$$\begin{aligned} H(X,Y) &= -\sum_{x} \sum_{y \in \Phi} p(x,y) \log(p(x,y)) & (3.50) \\ &= E\left[\log\left(\frac{1}{p(x,y)}\right)\right] & (3.51) \end{aligned}$$

The conditional entropy comes from the conditional probability distribution $p(y \mid x)$, which gives a measure of the average degree of uncertainty in Y for all possible

outcomes of X.

$$H(Y|X) = \sum_{x\in\Psi} p(x)H(Y \mid X = x) \quad = \quad -\sum_{x\in\Psi}\sum_{y\in\Phi} p(x,y)\log(p(y \mid x)) \quad (3.52)$$

We can now define a measure of the difference between two distributions called the called the *Kullback-Leibler Entropy* (1951), also called the *Kullback-Leibler Divergence* or *Kullback-Leibler Distance*. The Kullback-Leibler (K-L) entropy is a measure of how similar two distributions are to each other. Two distributions that are similar have small K-L distance. The K-L entropy $K(p,q)$ is similar to the distance between two distributions defined in the context of the Riemann metric in the space of distributions.

$$K(p,q) = \sum_{x\in\Psi} p(x)\log(\frac{p(x)}{q(x)}) \quad (3.53)$$

Although $K(p,q)$ carries the interpretation of the distance between two distributions, it is not a true distance because of the asymmetry between $p(x)$ and $q(x)$.

$$K(p,q) \neq K(q,p) \quad (3.54)$$

The K-L entropy is always positive and is zero if and only if $p(x) = q(x)$.

We can measure the statistical independence between two random variables X and Y by introducing the *mutual information* $I(X,Y)$. Let the random variables have distributions $p(x)$ and $q(x)$. Then the mutual information between X and Y is

$$\begin{aligned} I(X,Y) &= K(p(x,y), p(x)p(y)) \quad (3.55)\\ &= \sum_{x\in\Psi}\sum_{y\in\Phi} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \end{aligned}$$

The mutual information is symmetric so that $I(X,Y) = I(Y,X)$ and $I(X,X) = H(X)$. The mutual information measures the amount of information that Y conveys about X or vice-versa, and so it measures the degree of statistical correlation between X and Y. $I(X,Y)$ is zero if and only if X and Y are mutually independent.

Using set theory, it can be shown that the following are true

$$I(X,Y) = H(X) - H(X \mid Y) \tag{3.56}$$

$$I(X,Y) = H(Y) - H(Y \mid X) \tag{3.57}$$

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{3.58}$$

The preceding information theoretic definitions can be cast into forms of continuous random variables. For continuous random variable X with associated probability density function $f(x)$, the entropy is

$$h(X) = -\int_A f(x)\log(f(x))dx \tag{3.59}$$

where A is the domain of the continuous variable x. The K-L entropy of two continuous distributions $f(x)$ and $g(x)$ is defined analogously.

$$K(f,g) = \int f(x)\log\left(\frac{f(x)}{g(x)}\right)dx \tag{3.60}$$

Likewise, the mutual information between continuous variables X and Y is given by

$$I(X,Y) = \int f(x,y)\log\left(\frac{f(x,y)}{f(x)f(y)}\right)dxdy \tag{3.61}$$

We can define the complexity of a random vector as a measure of the interdependency of the different components. If we consider a p-variate distribution with joint density function $f(\mathbf{x}) = f(x_1, x_2, \ldots, x_p)$ and whose marginal densities are $f_j(x_j)$,

then we can write the information measure of dependence between random variables $x_1, x_2, \ldots, x_p$ as

$$\begin{aligned} I(\mathbf{x}) &= I(x_1, x_2, \ldots, x_p) \\ &= E\left[\log \frac{f(x_1, x_2, \ldots, x_p)}{f_1(x_1) f_2(x_2) \ldots f_p(x_p)}\right] \\ &= \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} f(x_1, x_2, \ldots, x_p) \log \frac{f(x_1, x_2, \ldots, x_p)}{f_1(x_1) f_2(x_2) \ldots f_p(x_p)} dx_{1 \ldots} dx_p \end{aligned} \tag{3.62}$$

Here, $I(\mathbf{x})$ is the K-L information divergence, which measures the expected dependencies between the component variables. This is also known as the expected information or mutual information. It can be shown that $I(\mathbf{x})$ is nonnegative. We can also note that if $f(x_1, x_2, \ldots, x_p) = f_1(x_1) f_2(x_2) \ldots f_p(x_p)$ for every p-tuple $(x_1, x_2, \ldots, x_p)$, then the the random variables $x_1, x_2, \ldots, x_p$ are mutually independent. In this case, then $\frac{f(x_1, x_2, \ldots, x_p)}{f_1(x_1) f_2(x_2) \ldots f_p(x_p)} = 1$, so the log in this case is zero. If $I(\mathbf{x}) > 0$, then there is some dependence between at least two of the variables.

We can find a relationship between the K-L divergence and Shannon's (1948) entropy by

$$\begin{aligned} I(\mathbf{x}) &= I(x_1, x_2, \ldots, x_p) \\ &= \sum_{j=1}^{p} H(x_j) - H(x_1, x_2, \ldots, x_p) \end{aligned} \tag{3.63}$$

where $H(x_j)$ is the marginal entropy and $H(x_1, x_2, \ldots, x_p)$ is the global or joint entropy

Let us define a multivariate normal density function $f(\mathbf{x})$ by

$$\begin{aligned} f(\mathbf{x}) &= f(x_1, x_2, \ldots, x_p) \\ &= (2\pi)^{-p/2} \mid \boldsymbol{\Sigma} \mid^{-p/2} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \end{aligned} \tag{3.64}$$

where $\boldsymbol{\mu}$=$(\mu_1, \ldots, \mu_p)$ and $\boldsymbol{\Sigma}$ is the covariance matrix.

We can write $\mathbf{x} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The joint entropy $H(\mathbf{x}) = H(x_1, x_2, \ldots, x_p)$ in the case where $\mu = 0$ is

$$\begin{aligned} H(\mathbf{x}) &= H(x_1, x_2, \ldots, x_p) \\ &= -\int f(\mathbf{x}) \log f(\mathbf{x}) d\mathbf{x} \\ &= \int f(\mathbf{x}) \left[\frac{p}{2} \log(2\pi) \mid \boldsymbol{\Sigma} \mid + \frac{1}{2} \mathbf{x}' \boldsymbol{\Sigma}^{-1} \mathbf{x} \right] d\mathbf{x} \\ &= \frac{p}{2} \log(2\pi) \mid \boldsymbol{\Sigma} \mid + \frac{1}{2} tr \left[\int f(\mathbf{x}) \boldsymbol{\Sigma}^{-1} \mathbf{x}' \mathbf{x} d\mathbf{x} \right] \end{aligned} \tag{3.65}$$

Note that $E[\mathbf{x}'\mathbf{x}] = \boldsymbol{\Sigma}$, so that

$$\begin{aligned} H(\mathbf{x}) &= H(x_1, x_2, \ldots, x_p) \\ &= \frac{p}{2} \log(2\pi) + \frac{p}{2} + \frac{1}{2} \log \mid \boldsymbol{\Sigma} \mid \\ &= \frac{p}{2} [\log(2\pi) + 1] + \frac{1}{2} \log \mid \boldsymbol{\Sigma} \mid \end{aligned} \tag{3.66}$$

The marginal entropy of a given variable is

$$\begin{aligned} H(x_j) &= -\int_{-\infty}^{\infty} f(x_j) \log f(x_j) dx_j \\ &= \frac{1}{2} \log(2\pi) + \frac{1}{2} + \frac{1}{2} \log(\sigma_j^2) \ , \ j = 1, \ldots, p \end{aligned} \tag{3.67}$$

3.6 Developing Information Complexity

We can now examine the complexity of a covariance matrix Σ for the multivariate normal distribution. Van Emden (1971) gave an initial definition of complexity. Substituting the expression for entropy gives

$$\begin{aligned} I(x_1, x_2, \ldots, x_p) &= \sum_{j=1}^{p} H(x_j) - H(x_1, x_2, \ldots, x_p) \\ &= \sum_{j=1}^{p} \left[\frac{1}{2} \log(2\pi) - \frac{1}{2} \log(\sigma_{jj}) + \frac{1}{2} \right] \\ &= \frac{p}{2} \log(2\pi) - \frac{1}{2} \log \mid \mathbf{\Sigma} \mid - \frac{p}{2} \end{aligned} \tag{3.68}$$

This expression reduces to

$$C_0(\mathbf{\Sigma}) = \frac{1}{2} \sum_{j=1}^{p} \log(\sigma_{jj}) - \frac{1}{2} \log \mid \mathbf{\Sigma} \mid \tag{3.69}$$

Here, $\sigma_{jj} = \sigma_j^2$ is the variance of the jth variable. This is also the jth diagonal element of $\mathbf{\Sigma}$. Van Emden (1971) demonstrates that the preceding result is not the best measure of complexity of covariance matrix $\mathbf{\Sigma}$ because $C_0(\mathbf{\Sigma})$ depends on the marginal distributions of the variables, and because the first term of $C_0(\mathbf{\Sigma})$ changes under orthonormal transformations. To improve on the initial definition $C_0(\mathbf{\Sigma})$, we can define the *maximal covariance complexity*. It can be shown (Van Emden 1971) that the maximal covariance complexity of $\mathbf{\Sigma}$ for a multivariate normal distribution is

$$\begin{aligned} C_1(\mathbf{\Sigma}) &= \max \left[H(x_1) + \ldots + H(x_p) - H(x_1, \ldots, x_p) \right] \\ &= \frac{p}{2} \log \left[\frac{tr(\mathbf{\Sigma})}{p} \right] - \frac{1}{2} \log \mid \mathbf{\Sigma} \mid \end{aligned} \tag{3.70}$$

where the maximum is evaluated over orthogonal transformation T of the overall coordinate systems $x_1, \ldots, x_p$. $C_1(\mathbf{\Sigma})$ measures both the inequalities between the variances and the contribution of the covariances in $\mathbf{\Sigma}$. This measure is independent of the coordinate system of the variances $\sigma_j^2,\ \ j = 1, \ldots, p$.

The complexity can be written as

$$C_1(\mathbf{\Sigma}) = \frac{1}{2} \log \frac{\left(\frac{tr(\Sigma)}{p}\right)^p}{\mid \mathbf{\Sigma} \mid} \tag{3.71}$$

which can be interpreted as the complexity between the geometric mean of the average total variation and the generalized variance, where $trace(\mathbf{\Sigma})$ is the total variation in the system and $\mid \mathbf{\Sigma} \mid$ is the expression for generalized variance. These quantities are measures of multivariate scatter. In general, the value of complexity is proportional to the amount of interaction among the variables. Large values of complexity reveal many interactions, while low values show less interactions.

3.7 Developing ICOMP for Linear Regression

We can show the theoretical development for ICOMP for a multiple regression model based on the complexity operation. We take the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.72}$$

where

$\mathbf{y}$ is a $(n \times 1)$ vector of values of the response value

$\mathbf{X}$ is a $(n \times q)$ is a model or design matrix with $rank(\mathbf{X}) = q = k + 1$

$\boldsymbol{\beta}$ is a $(q \times 1)$ vector of unknown coefficients

$\boldsymbol{\varepsilon}$ is a $(n \times 1)$ vector of random errors

We make the normal model assumption that $\boldsymbol{\varepsilon} \sim N(\mathbf{0},\sigma^2\mathbf{I})$. We know that the maximum likelihood estimates of $\boldsymbol{\beta}$ and σ^2 are

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} && (3.73) \\ \hat{\sigma}^2 &= s^2 && (3.74) \\ &= \frac{1}{n}(\mathbf{y}-\mathbf{y})'(\mathbf{y}-\mathbf{y}) \\ &= \frac{1}{n}\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} \end{aligned}$$

From properties of finite sampling, we have that

$$\hat{\boldsymbol{\beta}} \sim \mathbf{N}(\boldsymbol{\beta},\sigma^2(\mathbf{X}'\mathbf{X})^{-1}) \tag{3.75}$$

By the modeling assumptions, we know that the distribution of $\hat{\varepsilon}$ is multivariate normal with mean $E[\hat{\boldsymbol{\varepsilon}}] = \mathbf{0}$ and $Cov(\hat{\boldsymbol{\varepsilon}}) = \sigma^2(\mathbf{I}-\mathbf{H}) = \sigma^2(\mathbf{I}-\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X})$. Since σ^2 is an unknown parameter, it can be estimated by sample variance s^2, so we have

$$\text{Estimated } Cov(\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1} = s^2(\mathbf{X}'\mathbf{X})^{-1} \tag{3.76}$$

and

$$\text{Estimated } Cov(\hat{\boldsymbol{\varepsilon}}) = \hat{\sigma}^2(\mathbf{I}-\mathbf{H}) = s^2(\mathbf{I}-\mathbf{H}) \tag{3.77}$$

We can now derive ICOMP by using finite sampling distributions of parameter estimates. For the multivariate regression obeying the usual assumptions, ICOMP is defined by

$$ICOMP(\hat{\boldsymbol{\beta}},(\hat{\boldsymbol{\varepsilon}}|\hat{\boldsymbol{\beta}})) = -2\log L(\hat{\boldsymbol{\beta}},\hat{\sigma}^2) + 2[C_1(Cov(\hat{\boldsymbol{\beta}})) + C_1(Cov(\hat{\boldsymbol{\varepsilon}}))] \tag{3.78}$$

This expression for ICOMP has the following components:

1. The first component comes from maximizing the log-likelihood of the model. As previously shown, we can express this as

$$-2\log L(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n \tag{3.79}$$

where $\hat{\sigma}^2 = \frac{SS_{res}}{n}$

2. The second component is the complexity of the $\hat{\beta}$ term

$$\begin{aligned} C_1(Cov(\hat{\boldsymbol{\beta}})) &= \frac{q}{2}\log\left[\frac{tr(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1})}{q}\right] - \frac{1}{2}\log\left[\det(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1})\right] \\ &= \dot{C}_1((\mathbf{X}'\mathbf{X})^{-1}) \end{aligned}$$

Here, $q = rank((\mathbf{X}'\mathbf{X})^{-1})$

Because $\dot{C}_1(\cdots)$ is scale invariant, $\hat{\sigma}^2$ can be factored out

3. For the third component, let

$$\mathbf{M} = (\mathbf{I} - \mathbf{H}) = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \tag{3.80}$$

It can be shown that, under normal modeling assumptions $\mathbf{M} = (\mathbf{I} - \mathbf{H})$

$$C_1^*(\mathbf{M}) = 0 \tag{3.81}$$

As long as $\varepsilon \sim N(\mathbf{0}, \hat{\sigma}^2\mathbf{I})$, then the third component of ICOMP will equal 0.

We can therefore write ICOMP for multivariate regression as

$$ICOMP(REG) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2\dot{C}_1((\mathbf{X}'\mathbf{X})^{-1}) \tag{3.82}$$

In addition, we can also derive ICOMP by using the inverse of the Fisher information matrix. For the multivariate regression model previously given

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{3.83}$$

we have the covariance matrix equal to the inverse Fisher information matrix of

$$Cov(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) = \mathbf{F}^{-1} = \begin{bmatrix} \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1} & \mathbf{0} \\ \mathbf{0}' & \frac{2\hat{\sigma}^2}{n} \end{bmatrix} \tag{3.84}$$

We can again consider ICOMP as composed of several parts. The first part is the log-likelihood evaluated at the maxima points, and the second component is the complexity of the inverse Fisher information matrix.

$$\begin{aligned} ICOMP(IFIM) &= -2\log L(\hat{\boldsymbol{\beta}},\hat{\sigma}^2) + 2C_1(\mathbf{F}^{-1}) \\ &= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + \\ &\quad (q+1)\log\left[\frac{tr(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}) + \frac{2\hat{\sigma}^4}{n}}{q+1}\right] \\ &\quad -\log\left(\det\left(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}\right)\right) - \log\left(\frac{2\hat{\sigma}^4}{n}\right) \end{aligned} \tag{3.85}$$

We can again factor out $\hat{\sigma}^2$ from the complexity term since it is scale invariant, which yields

$$\begin{aligned} ICOMP(IFIM) &= n\log(2\pi) + n\log(\hat{\sigma}^2) + n + \\ &\quad (q+1)\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1}) + \frac{2\hat{\sigma}^2}{n}}{q+1}\right] - \\ &\quad \log\left(\det\left((\mathbf{X}'\mathbf{X})^{-1}\right)\right) - \log\left(\frac{2\hat{\sigma}^2}{n}\right) \end{aligned} \tag{3.86}$$

Further simplification yields the expression

$$\begin{aligned} ICOMP(IFIM) &= n\log(2\pi) + (n - \frac{1}{2})\log(\hat{\sigma}^2) + n \\ &\quad +(q+1)\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1}) + \frac{2\hat{\sigma}^2}{n}}{q+1}\right] \\ &\quad -\log\left(\det\left((\mathbf{X}'\mathbf{X})^{-1}\right)\right) - \log\left(\frac{n}{2}\right) \end{aligned} \tag{3.87}$$

3.8 Conclusion

This concludes our derivation of ICOMP and an overview of information based statistics. In summary, we can say that information based statistical methods removes the ambiguity and subjectivity found in classical statistics, and provides a framework for statistical analysis that is more general than that of classical statistics. For more information about the development of information scored regression, the interested reader can consult the text edited by Bozdogan (2004).

Chapter 4

Modeling with Genetic Algorithms

4.1 Introduction

Genetic Algorithms are optimization methods. They form a branch of evolutionary computation and contribute to the field of Artificial Intelligence. The traditional Genetic Algorithm (GA) paradigm represents possible solutions to a problem by binary strings. The collection of strings, called the population, uses principles of Darwinian evolution to mimic organisms in an environment adapting to changing environmental conditions, with members of the population mating and reproducing. The probability of reproduction is proportional to how well the population member solves the problem, giving the most well-adapted organisms the highest probability of reproduction. Through many generations of this process, members evolve optimal solutions. By cleverly implementing GA's, researchers can find solutions to complex, sometimes previously intractable, problems. GA's are especially well-suited to problems with vast, nonlinear search spaces where gradient based algorithms can become trapped in local optima. GA's do not rely on local slope calculations, but instead, use GA operations to globally search the solution space.

Because GA's are efficient searching algorithms, they can be implemented as a tool for statistical model selection. Modern data analysis must process large numbers of variables. Researchers must implement effective methods to model complex multivariate data. While classical statistical methods use stepwise regression to select the best model, ambiguities in model selection probabilities and levels of significance grows with the number of model variables. Classical regression methods try to minimize the sum of squared error between observed data and a statistical model. This approach generally overfits multivariate regression models. In addition, according the published literature (Sokal and Rolf 1981, pg 668), stepwise regression cannot adequately search highly multivariate parameter space and it generally calculates suboptimal models.

Since Akaike (1973) introduced information scoring as a way to select parsimonious models, the field of information scored statistics has continued to expand. Information based statistics assigns scores to different combinations of model parameters, trying to find a balance between Lack of Fit and Lack of Parsimony. Akaike Information Criteria (AIC) was the first example of a regression scoring function. ICOMP (Bozdogan 1988, 1990a) is a more modern measure of regression model quality. Information scored regression regards the combination of variables that achieves the lowest score as the best for describing the observed data set. However, in order to find the best model score, we may have to process a large number of model combinations. The number of possible models for k variables is $2^k - 1$. For a relatively small number of variables (less than approximately 15), researchers can calculate every combination of variables to find the best model on a common computer. For a larger number of variables, this subsetting scheme becomes impractical because the number of variable combinations that must be evaluated doubles for each added variable. To overcome this handicap, Bearse and Bozdogan (2002) implemented GA's in model

selection, using ICOMP as the measure model quality. In this way, the GA acts as a computational shortcut. Instead of scoring every possible combination of model parameters, the GA samples a small subset of all of the model combinations. The best model, which is the parameter combination with the smallest ICOMP score, naturally evolves in the GA framework. This innovation combines the discrimination ability of information based statistics with the efficiency of the GA data structure.

During the 1950's and 1960's, a number of computer scientists considered how optimization problems in engineering could be tackled by implementing programs that simulate biological evolution. In these simulations, a population of candidate solutions could interact and reproduce, yielding optimal results to complex problems. Evolutionary computing was introduced in the 1970's by Rechenberg (1973) in his work "Evolution strategies" (Evolutionsstrategie in original form). Continuing this work, the modern form of GA's was invented by John Holland (1975) and his students and colleagues. This lead to Holland's 1975 book "Adaptation in Natural and Artificial Systems." Since then, GA's have been widely studied and applied in many fields of science and engineering. Not only do GA's provide alternative methods to solving problems, but they also consistently outperform other methods used in searching highly nonlinear spaces in terms of speed and efficiency. Researchers have demonstrated that many of the real world problems that involve finding optimal combinations of parameters which might prove difficult for traditional methods are ideal for GA's.

The traditional way of approaching optimization problems is to use the gradient ascent/decent method. In a gradient based approach, the researcher gives some educated guess about the optimal values of a system. The program then iteratively adjusts these initial values according to the local gradient of the "parameter landscape." When the values stop changing significantly or the gradient is near zero,

this algorithm terminates at an optimal point. By contrast, a GA does not rely on gradient calculations or initial starting points. Each of the strings in the GA population independently tries to find the maximum value of the search space, creating an implicitly parallel search process. Consequently, the GA has a much higher probability of finding a global maximum because its search process is driven by competition among the members and, unlike a gradient based algorithm, does not generally get stuck in local optima.

A closely related concept to GA optimization is the *fitness landscape* of a problem. The biologist Sewell Wright (1931) defined the term fitness landscape to describe the representation of how biological organisms adapt to an environment. This landscape has peaks, valleys, ridges, and similar landscape terrain. According to Wright, evolution forces succeeding generations of organisms to move along the fitness landscape in such a way that they try to find local peaks. This can explain why different species that had a common ancestor diverge. As different populations of the progenitor specie migrated to different environments, each group tried to find a local maximum in the fitness landscape. As the different populations became more isolated, the populations became stuck in their own local peaks and lost contact with the other populations, causing them to evolve into different organisms.

The computational analogy of this biological concept causes the GA to adapt to changes within its own fitness landscape. If a researcher can cast a problem into a framework where different parameter combinations can be represented as genetic strings, and there is a way to translate the quality of the solution into a kind of fitness representation, then the researcher can employ the GA as an accurate and efficient way to find solutions to these kinds of complex problems.

This chapter will outline the development of GA's by introducing the theory and terminology. The chapter will then demonstrate how researchers can structure regression routines as GA's using ICOMP as a measure of model quality.

4.2 Genetic Algorithm Theory

In order to implement a GA in numerical optimization problems, we must invent a way to convert the variables to be optimized into representation of GA *chromosomes*. In the traditional GA representation, chromosomes are strings of binary numbers. The set of chromosomes that the algorithm uses is called the *population* and each position in the string that can assume a value of 0 or 1 is a *gene*. One of the advantages of the GA optimization process is its ability to find optimal solutions using only a small subset of possible gene combinations in the search space.

The GA optimizes the solution of a problem by allowing the population of chromosomes to interact with each other, exchange genetic information and reproduce. The strategy used in optimization is analogous to that employed by species trying to maximize their position in a fitness landscape. The algorithm must incorporate a *fitness function* that assigns a score to each chromosome based on its ability solve the problem under consideration. At each iteration of the GA process, the chromosomes in the population are ranked according to their fitness.

The definition of a fitness function is problem-specific with no general guidelines or methods. The fitness can be normalized to relative values or unnormalized natural values, and the quality can be increasing or decreasing. In a business management situation, the fitness function can be the amount of predicted profits given a possible set of business conditions. The fitness function in this case is obviously maximized. In an aerospace engineering firm, variables related to aircraft construction could be

chosen to minimize weight. The usefulness of the GA lies in the clever definition of a fitness function applied to some optimization context.

4.3 Implementing Binary Genetic Algorithms

In order to use a binary GA in problem solving, the researcher must find a way to represent possible solutions to a problem by binary bits. The GA process starts with a randomly initialized population of binary chromosomes. The GA loop then forces chromosomes to evolve using the GA operations of *selection*, *crossover*, and *mutation*. We shall describe each of these operations in turn.

Selection is the when the program chooses high-ranking chromosomes for reproduction. At each iteration, chromosomes are ranked by the fitness function. Those chromosomes with good fitness values are given a higher probability of reproduction than those with poorer fitness values. There are a number of strategies for using a selection operation, most of which involve some proportionality between fitness and reproduction probability. A popular selection method is the *roulette wheel sampling* (Goldberg 1989) which is analogous to assigning ranked probabilities to proportionate areas of a roulette wheel. Using this method, the relative probabilities of the chromosomes are mapped onto a probability distribution, with the highest-ranking probabilities assigned the largest probability. The algorithm is iterated until a single chromosome dominates the population or the ranks of the chromosomes do not change over several iterations. Some researchers plot the average fitness and the best fitness as the iterations progress, tracking the convergence of population. These plots generally appear as exponentially increasing or decreasing functions that asymptotically approach an optimum value. In addition, some researchers advocate elitism in

the reproduction operation. An elitist reproductive strategy forces the best ranking members of each generation into subsequent generations.

Some researchers elect to kill the lowest ranking members of the population of chromosomes at each iteration of the algorithm. While this *greedy selection* speeds the convergence of the algorithm, there can be loss of valuable genetic material if regular mass extinctions occur in the population. Many researchers prefer a modest extinction rate of perhaps 10% balanced by the same amount of randomly reinitialized chromosomes. If a new random chromosome happens to be close to an optimal solution, then it will naturally migrate to one of the highest-ranking slots at the next iteration.

The crossover operation allows chromosomes to exchange genetic information, producing two offspring chromosomes. In this operation, two chromosomes are randomly selected for mating, and then the crossover point within these chromosomes is randomly selected. The parts of the binary strings after the crossover point are exchanged between the parent chromosomes to produce offspring chromosomes. For example, suppose that the two parent chromosomes are

010111010

111110100

Suppose also that the fourth gene is selected as the crossover point, so that genetic material after the fifth gene is exchanged. The resulting offspring would be

010110100

111111010

The crossover operation drives the chromosomes to produce high-ranking offspring. This is especially true of pairs of chromosomes that are already quite fit. The offspring of two highly fit chromosomes can produce the best combination of genes, yielding a child that is the optimal solution. A theory of GA's called the *building block hypothesis* states that substrings of chromosomes that individually contribute to an optimal solution can combine through crossover to build the best string (Mitchell 1998). Some judgement is required in selecting a crossover rate. Too high of a crossover rate tends disassociate too many highly-fit chromosomes, resulting in a population that never converges. Many researchers consider a crossover rate of 5% to 10% adequate to force the population to evolve optimal solutions.

Mutation produces changes in gene sequences by randomly turning a 1 into a 0 and vice-versa. Mutation is the most passive GA operation because it only produces minor changes the population. Many researchers agree that a small mutation probability of between 1% and 5% suffices to randomly vary highly-ranked chromosomes. Mutation can be regarded as "fine tuning" the fittest chromosomes to yield the best solutions to complex problems.

The GA process can be summarized as follows. A fitness function is used to evaluate candidate solutions, and reproductive success varies with fitness. The GA then incorporates an iterative loop as shown (Mitchell 1998).

1. Generate an initial population M_0.

2. Compute the fitness $u(m)$ for each individual m in the current population M_t.

3. Define selection probabilities $p(m)$ for each individual m in M_t so that $p(m)$ is proportional to $u(m)$.

4. Generate M_{t+1} by probabilistically selecting individuals from M_t to produce offspring via genetic operators.

5. Repeat step 2 until a satisfying solution is obtained.

4.4 Binary GA Used in Statistical Modeling

We can now turn our attention to applying GA's to multivariate linear regression. Selecting the best regressor variables that summarize and describe a data set is the main goal of statistical modeling. This process progressively becomes more difficult as the possible number of variables grows. Information based modeling provides a good framework in which to implement the GA for model selection.

Bearse and Bozdogan (2002) made an insightful analogy between linear models and GA chromosomes. A binary GA string can represent the inclusion or exclusion of variables from a linear regression model. We can demonstrate this correspondence as follows. A linear regression model is represented as a linear equation as:

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_j \end{aligned} \tag{4.1}$$

The GA string can represent which variables are included in the current model, with 1 representing inclusion and 0 representing exclusion. For example, suppose that there are 5 possible variables that can be used to build a regression model. Then there are 2^5 possible combinations of models. These range from only the constant term

$$y_i = \beta_0 \tag{4.2}$$

to the saturated model that contains all variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \tag{4.3}$$

There is a one-to-one correspondence between every possible chromosome of length 5 and every possible linear model that can be constructed with 5 variables. Take some examples

$$y = \beta_0 + \beta_1 x_1 + \beta_4 x_4 + \beta_5 x_5 \qquad 10011$$

$$y = \beta_0 + \beta_2 x_2 + \beta_4 x_4 \qquad 01010$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \qquad 11111$$

$$y = \beta_0 + \beta_2 x_2 + \beta_3 x_3 \qquad 01100$$

This analogy between variables included in the model and bits in binary strings can be extended to a large numbers of variables. In fact, there is no theoretical limit to the number of variables that can be represented by binary strings.

In order to rank the quality of candidate solutions, the GA needs some fitness function. The information measure of complexity ICOMP qualifies as a fitness function because it gives a score to models based on their ability to describe a data set. Moreover, the smaller the ICOMP score, the better a model describes the data, so we can employ ICOMP as a measure of fitness by minimizing ICOMP, or equivalently, maximizing the negative of ICOMP. For data whose errors are normally distributed, we can use ICOMP derived from the finite sampling properties or ICOMP derived from the inverse Fisher information matrix. That is, assuming that

$$\epsilon \sim N(\mathbf{0}, \sigma^2 \mathbf{I}) \tag{4.4}$$

we can write the ICOMP based on finite sampling as

$$ICOMP(REG) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2C_1((\mathbf{X}'\mathbf{X})^{-1}) \tag{4.5}$$

with

$$C_1((\mathbf{X}'\mathbf{X})^{-1}) = \frac{q}{2}\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1})}{q}\right] - \frac{1}{2}\log\left[\det((\mathbf{X}'\mathbf{X})^{-1})\right] \tag{4.6}$$

Here, $q = rank((\mathbf{X}'\mathbf{X})^{-1})$. Likewise, under the assumption of normality, the Inverse Fisher information matrix is

$$Cov(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) = \mathbf{F}^{-1} = \begin{bmatrix} \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1} & \mathbf{0} \\ \mathbf{0}' & \frac{2\hat{\sigma}^2}{n} \end{bmatrix} \tag{4.7}$$

Using this expression, we can derive the Inverse Fisher Information Matrix version of ICOMP as

$$\begin{aligned} ICOMP(IFIM) &= n\log(2\pi) + (n - \frac{1}{2})\log(\hat{\sigma}^2) + n \\ &\quad + (q+1)\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1}) + \frac{2\hat{\sigma}^2}{n}}{q+1}\right] \\ &\quad - \log\left(\det\left((\mathbf{X}'\mathbf{X})^{-1}\right)\right) - \log\left(\frac{n}{2}\right) \end{aligned} \tag{4.8}$$

Published literature (Bozdogan 2004) demonstrates that ICOMP achieves the most parsimonious model by controlling the risk of insufficient and overparameterized models. The complexity term balances these competing forces and guards against collinearity.

A minor point regarding the number of possible models is that in the standard GA population, all possible set decompositions are possible, including the empty set. However, ICOMP suffers a singularity when evaluating the empty set model, and regression modelers generally do not consider it as a viable solution. Therefore, in our GA implementation, if the GA generates an empty set (all zeros) solution, we randomly mutate one of the entries to give it a non empty solution.

The strategy of implementing a GA in a regression modeling context is as follows.

1. The algorithm randomly initializes a population of chromosomes that represents possible models.

2. The algorithm applies mutation and recombination operations to the population.

3. The algorithm then evaluates ICOMP for each chromosome, thereby assigning it a rank.

4. It gives models that have better (lower) ICOMP scores a higher probability of reproduction than those with a poorer ICOMP score.

5. Continue this GA the loop until it converges to a dominate member of the population.

4.5 Conclusion

This concludes our discussion of the theory of binary GA's. We introduced ideas behind GA operation and their related terminology. The interested reader can find more discussion in the texts by Mitchell (1998) and Koza (1992). For more information specifically about using GA's in linear regression models, the reader can consult Bozdogan (2004).

Chapter 5

Molecular Spectroscopy Theory

5.1 Introduction

Molecular vibration-rotation spectroscopy infers properties of molecules from studying their spectra. Molecules generally absorb and emit vibrational and rotational energy in the microwave and infrared regions of the electromagnetic spectrum. Researchers derive many properties of molecules from studying their spectra, including isotopic masses and molecular structure. In addition, by comparing catalogs of laboratory spectra with spectra observed by terrestrial or astronomical observations, researchers can derive temperatures, pressures, and relative abundances of molecular species observed to be at remote locations.

The study of infrared spectroscopy can be traced back to a publication from 1800 when Sir William Hershel observed that thermometers measured higher temperatures near the red end of the spectrum, indicating that some unknown radiation transmitted heat to the thermometer. In 1840, Sir John Hershel (son of William) observed that black alcohol-soaked paper dried more quickly when exposed to certain spectral regions. Many historical names contributed to the development of infrared spectroscopy,

including Kirchhoff, Bunsen, Coblentz, Wood, Pfund, and many others. Later, 1942 saw the development of modern automatically-recording prism spectrometers, which grew from the wartime campaign to construct infrared detectors. Subsequently, scientists recorded and studied numerous spectra of organic and inorganic compounds (Blass and Nielsen 1974).

The energies generated by motions of molecules can range from 4000 Angstroms to 1000 microns. Energies arising from electronic motions of molecules generally fall in the ultraviolet and visible regions. If the emitted energy comes from molecular vibrations, it will range from about 1 micron to 25 microns, while molecular rotations generate energy with wavelengths longer than 25 microns (Blass and Nielsen 1974).

This chapter will review the background development of molecular vibration-rotation theory. Starting from the analytically simple diatomic molecule, it will then generalize the derived expression to more complex molecules. The end of the chapter will show how the derived complex expressions can be used in least-squares regression to calculate molecular constant values.

5.2 Diatomic Molecular Vibration

As a first example of applying analytical vibration-rotation equations to a molecule, we will examine the development of a diatomic molecule. This is a good paradigm for the development of vibration-rotation equations because it is generalizable to more complex situations. We will first examine the vibration and rotation dynamics separately before we consider the vibration-rotation interactions.

Suppose that two atoms are distance r apart and that the force between them is the algebraic sum of attractive and repulsive electromagnetic forces. Let the equilibrium distance between the two atoms be r_e. The potential energy function approaches

infinity as $r \to 0$, and it approaches a fixed value as $r \to \infty$. The energy required to move the nuclei from $r = r_e$ to $r = \infty$ is the dissociation energy of the molecule (Blass and Nielsen 1974).

Because we are considering small amplitude vibrations of the molecule, a good approximation of the restoring force between the nuclei is a Hooke's law force, with the force taken as proportional to displacement from equilibrium. Therefore, the potential energy takes the form $V = \frac{1}{2}kx^2$, where k is a force constant which is connected to the frequency of vibration by

$$\omega_e = \left(\frac{1}{2}\pi c\right)\frac{\sqrt{k}}{\sqrt{\mu}} \tag{5.1}$$

where ω_e is the frequency of small amplitude vibrations and μ is the reduced mass with the form

$$\mu = \frac{m_1 m_2}{m_1 + m_2} \tag{5.2}$$

As a result of this definition, we can see that changing the atomic mass of a component of the molecule changes the corresponding vibration frequency. Hence, from this property, isotopic dependencies of vibration frequencies can be derived.

While the Hooke's law potential is a good starting point for modeling atomic interaction forces, the study of molecular spectra shows that more complex potential functions better model real molecules. We can expand the potential function as a Taylor series (Blass and Nielsen 1974)

$$H = \frac{1}{2}kx^2 + k_{111}x^3 + k_{1111}x^4 + \ldots \tag{5.3}$$

Turning to the quantum mechanical description of the molecular vibration problem, the Hamiltonian for a diatomic molecule becomes

$$H = -\frac{\hbar^2}{2m_1}\frac{d^2}{dx_1^2} - \frac{\hbar^2}{2m_2}\frac{d^2}{dx_2^2} + \frac{1}{2}kx^2 \tag{5.4}$$

or defining effective mass μ gives

$$H = -\frac{\hbar^2}{2\mu}\frac{d^2}{dx^2} + \frac{1}{2}kx^2 \tag{5.5}$$

We can see that, since this expression of the Hamiltonian is in the form of a harmonic oscillator, solutions have energies of

$$E_\nu = \left(\nu + \frac{1}{2}\right)\hbar\omega \tag{5.6}$$

where $\omega = \left(\frac{k}{\mu}\right)^{1/2}$ and $\nu = 0, 1, 2, \ldots$. The energies form a ladder of possible energy states, with step separation $\hbar\omega$, while the corresponding wavefunctions have Gaussian shapes multiplied by Hermite polynomials (Blass and Nielsen 1974).

As in the classical case, a more realistic quantum mechanical description of the molecular vibrator must include anharmonic terms in the potential energy. The potential can be expanded in a Taylor series. The corresponding Schodinger equation must then be solved to give a description of the molecular wavefunctions. This process must use numerical solution techniques because no analytical solutions can be derived.

The vibration selection rule implies that the transition matrix element is

$$\mu_{\nu\nu'} = \langle \nu \mid \mu \mid \nu' \rangle \tag{5.7}$$

We can express the change in dipole moment from equilibrium as:

$$\mu = \mu_0 + \left(\frac{d\mu}{dx}\right)_0 x + \frac{1}{2}\left(\frac{d^2\mu}{dx^2}\right)_0 x^2 + \ldots \tag{5.8}$$

The transition matrix element then becomes:

$$\begin{aligned} \mu_{\nu\nu'} &= \mu_0\langle\nu \mid \nu'\rangle + \left(\frac{d\mu}{dx}\right)_0 \langle\nu \mid x \mid \nu'\rangle + \frac{1}{2}\left(\frac{d^2\mu}{dx^2}\right)_0 \langle\nu \mid x^2 \mid \nu'\rangle + \ldots \\ &= \left(\frac{d\mu}{dx}\right)_0 \langle\nu \mid x \mid \nu'\rangle + \frac{1}{2}\left(\frac{d^2\mu}{dx^2}\right)_0 \langle\nu \mid x^2 \mid \nu'\rangle + \ldots \end{aligned} \tag{5.9}$$

For small amplitude oscillations in the harmonic potential, we have

$$\mu_{\nu\nu'} \approx \left(\frac{d\mu}{dx}\right)_0 \langle\nu \mid x \mid \nu'\rangle \tag{5.10}$$

This leads to the selection rule for electric dipole transitions for harmonic vibrations:

$$\Delta\nu = \pm 1 \tag{5.11}$$

When it is necessary to include anharmonicity terms in the potential function, higher order terms must be included in the transition matrix. Therefore, the selection rule becomes

$$\Delta\nu = \pm 2 \tag{5.12}$$

for x^2 terms and so on for higher order contributions. These electrical anharmonicities allow transitions from the ground vibrational state to higher overtone states.

5.3 Diatomic Molecular Rotation

In addition to considering molecular vibrations, we would like to analyze the dynamics of molecular rotation. We can again consider a diatomic molecule as a good paradigm for starting the analysis. First, consider the molecule as a nonvibrating rigid rotor. If the molecule has reduced mass μ and equilibrium distance r_e, then the classical kinetic energy of the molecule is $E_R = \frac{P^2}{2I_c}$, where P is the total angular momentum and $I_e = \mu r_e^2$ is the moment of inertia.

When we transform to the quantum mechanical description of the rotating diatomic molecule, we can replace P^2 by $\frac{J(J+1)h^2}{2\pi}$, where J takes values $0, 1, 2, ...$ The rotational energy is usually expressed as

$$\frac{E_R}{hc}(cm^{-1}) = J(J+1)B_e \tag{5.13}$$

B_e is the equilibrium rotational constant defined as

$$B_e = \frac{h}{8\pi^2 I_e c} \tag{5.14}$$

It can be shown that the dipole selection rule is $\Delta J = \pm 1$. These properties cause pure rotational transitions to appear at predictable intervals of 0, $2B_e$, $6B_e$, $12B_e$, etc. (Blass and Nielsen 1974).

While the rigid rotator is a good first approximation to a rotating diatomic molecule, in reality, the atoms experience centrifugal forces that stretch the bonds. These centrifugal forces increase the average internuclear separation. Analysts account for this interaction by adding a centrifugal distortion constant. The rotational energy for

the $\nu = 0$ vibrational state then becomes:

$$\begin{aligned} \frac{E_R}{hc}(cm^{-1}) &= \left[B_e - \frac{1}{2}\alpha - \left(D_e - \frac{1}{2}\beta\right) J\left(J+1\right)\right]\left(J+1\right) \\ &= B_0 J\left(J+1\right) - D_0 J^2\left(J+1\right)^2 \end{aligned} \tag{5.15}$$

where B_0 is the ground vibrational state rotational constant and D_0 is the centrifugal distortion constant. Hence, in some vibrational state ν, the associated energy levels are

$$\frac{E_R}{hc}(cm^{-1}) = B_\nu J(J+1) - D_\nu J^2(J+1)^2 \tag{5.16}$$

with

$$B_\nu = B_e - \alpha(v + \frac{1}{2}) \tag{5.17}$$

$$D_\nu = D_e - \beta(v + \frac{1}{2}) \tag{5.18}$$

Further corrections can be added in the same way that include $J^3(J+1)^3$ terms (Blass and Nielsen 1974).

5.4 Diatomic Molecular Vibration-Rotation

We can combine the anharmonic vibrator with a nonrigid rotator. Classical mechanical considerations predict absorption frequencies at $\omega_e \pm \omega_R$. Combining the quantum mechanical expressions for energy gives

$$\begin{aligned} E_{VR} &= \omega_e(\nu + \frac{1}{2}) - x_e\omega_e\left(\nu + \frac{1}{2}\right)^2 + y_e\omega_e\left(\nu + \frac{1}{2}\right)^3 + \ldots + \\ &\quad B_\nu J(J+1) - D_\nu J^2(J+1)^2 + H_\nu J^3(J+1)^3 \end{aligned} \tag{5.19}$$

where

$$\begin{aligned} B_\nu &= B_e - \alpha(\nu + \frac{1}{2}) + \gamma(\nu + \frac{1}{2})^2 \\ &= B_0 - \alpha\nu + \gamma(\nu + \nu^2) \end{aligned} \tag{5.20}$$

$$\begin{aligned} D_\nu &= D_e - \beta(\nu + \frac{1}{2}) \\ &= D_e - \beta\nu \end{aligned} \tag{5.21}$$

$$\begin{aligned} H_\nu &= H_e - \delta\left(\nu + \frac{1}{2}\right) \\ &= H_0 - \delta\nu \end{aligned} \tag{5.22}$$

We can now consider vibration-rotation transitions as transitions from the ground state, so that the observed energies will be the difference between the ground state and an excited state. Hence, we can express the energy differences as $E_{VR}(\Delta\nu, J + \Delta J) - E_{VR}(0, J) = \Delta E_{VR}$

$$\begin{aligned} \Delta E_{VR} = {} & \omega_e\Delta\nu - x_e\omega_e\left(\Delta\nu + 1\right)\Delta\nu + \\ & y_e\omega_e\left(\Delta\nu + \frac{3}{2}\Delta\nu + \frac{3}{4}\Delta\nu\right)\Delta\nu + \ldots + \\ & B_0\left(2J + 1 + \Delta J\right)\Delta J - \alpha\Delta\nu(J + \Delta J)(J + 1 + \Delta J) + \\ & \gamma(\Delta\nu + 1)\Delta\nu\left(J + \Delta J\right)\left(J + 1 + \Delta J\right) - \\ & D_0\left[\left(J + \Delta J\right)^2\left(J + 1 + \Delta J\right)^2 - J^2\left(J + 1\right)^2\right] + \\ & \beta\Delta\nu\left(J + \Delta J\right)^2\left(J + 1 + \Delta J\right)^2 + \\ & H_0\left[\left(J + \Delta J\right)^3\left(J + 1 + \Delta J\right)^3 - J^3\left(J + 1\right)^3\right] \end{aligned} \tag{5.23}$$

Transitions for which $\Delta J = 1$ lead to the R-branch of the spectrum while transitions where $\Delta J = -1$ gives rise to the P-branch. Identifying these spectral components is an important step in analyzing the data. Furthermore, the energy equations become a starting point for a least squares treatment of the spectral data in order to recover numerical parameter estimates for a given molecule.

5.5 Polyatomic Molecular Rotation

The theory of molecular rotation begins with defining the moment of inertia. We can write the moment of inertia of a molecule rotating about axis q as

$$I_{qq} = \sum_i m_i x_i^2(q) \tag{5.24}$$

where $x_i(q)$ is the perpendicular distance of the atom i with mass m_i from axis q. Using this definition, we can write the classical kinetic energy of the rotating molecule as

$$T = \frac{1}{2} \sum_q I_{qq} \omega_q^2 = \sum_q \frac{J_q^2}{2I_{qq}} \tag{5.25}$$

where ω_q is the angular frequency about axis q. If there is no potential term in this context, then the Hamiltonian is the sum of the rotational kinetic energies of the molecule.

$$T = \frac{J_x^2}{2I_{xx}} + \frac{J_y^2}{2I_{yy}} + \frac{J_z^2}{2I_{zz}} \tag{5.26}$$

We can specialize this expression to specific types of molecules. A symmetric rotor has a symmetry axis where two of the moments of inertia are equal. Let $I_\perp = I_{xx} = I_{yy}$ and let $I_\parallel = I_{zz}$. We then have

$$H = \frac{J_x^2 + J_y^2}{2I_\perp} + \frac{J_z^2}{2I_\parallel} \tag{5.27}$$

We can rewrite this Hamiltonian in terms of the magnitude of the total angular momentum $J^2 = J_x^2 + J_y^2 + J_z^2$ as

$$H = \frac{J^2}{2I_\perp} + \left(\frac{1}{2I_\parallel} + \frac{1}{2I_\perp}\right) J_z^2 \tag{5.28}$$

In a quantum mechanical situation, we regard J^2 and J_z as operators with corresponding eigenvalues. Letting K represent the quantum number of angular momentum of the internal symmetry axis of the molecule and M_J be the quantum number of the component of angular momentum in the laboratory's z-axis, we have

$$E\left(J, K, M_J\right) = \frac{J\left(J+1\right)}{2I_\perp}\hbar^2 + \left(\frac{1}{2I_\parallel} + \frac{1}{2I_\perp}\right) K^2\hbar^2 \tag{5.29}$$

Here, the range of values for the quantum numbers are

$$J = 0, 1, 2, \ldots \tag{5.30}$$

$$K = J, J-1, \ldots, -J \tag{5.31}$$

$$M_J = J, J-1, \ldots, -J \tag{5.32}$$

Although the quantum number M_J does not explicitly appear in the energy expression, we must include it in order to have a complete description of the energy state of the molecule. We can understand this in the sense that, if there is no external field applied to the molecule, then there is no preferred direction for the molecule to orient itself. The quantum number K describes the distribution of angular momenta over the molecule. If $|K| \approx J$, then almost all of the molecule's angular momenta is around its symmetry axis, while if $|K| \approx 0$, almost all of the molecule's angular momentum is around an axis perpendicular to the symmetry axis. We can also observe that since the energy depends on K^2, the energy does not depend on the direction of rotation

around the symmetry axis. This is consistent with the physical interpretation of the quantum numbers.

We can also define rotational constants A and B to be

$$A = \frac{\hbar}{4\pi c I_{\|}} \tag{5.33}$$

$$B = \frac{\hbar}{4\pi c I_{\perp}} \tag{5.34}$$

Then the energy E is related to transition frequency $F(cm^{-1})$ as

$$E\left(J, K, M_J\right) = hcF(J, K, M_J) \tag{5.35}$$

so we can write

$$F(J, K, M_J) = BJ(J+1) + (A-B)K^2 \tag{5.36}$$

Each K level with $K \neq 0$ is $2\left(2J+1\right)$ fold degenerate because each M_J can assume $2(J+1)$ different values for a given value of J. If $K = 0$, then the degeneracy of M_J is $2J+1$ because K takes only a single value. Limiting cases of the energy are as follows. For $K = 0$, the energy is

$$F(J, 0, M_J) = BJ(J+1) \tag{5.37}$$

This is the situation where all of the kinetic energy comes from the molecule rotating about an axis perpendicular to the symmetry axis. For the maximum K value of $|K| = J$, the energy is

$$F(J, \pm J, M_J) = AJ^2 + BJ \tag{5.38}$$

Here, most of the energy comes from rotation about the symmetry axis.

Some special cases arise from the energy expression. A spherical rotor is one where all three moments of inertia are equal. In this case, $A = B$ so the frequency expression becomes

$$F\left(J, K, M_J\right) = BJ(J+1) \tag{5.39}$$

Here, the transition frequency shows no dependency on both K and M_J. This agrees with our concept of spherical symmetry in which no direction is preferred over any other. However, each level is now $(2J+1)^2$ degenerate because of the multitude of states available for K and M_J for each value of J.

A linear rotor is one that has only two moments of inertia. In this case, $K \equiv 0$, so the transition equation becomes

$$F(J, M_J) = BJ(J+1) \tag{5.40}$$

and shows no K dependency. The degeneracy for each level is $2(J+1)$ because the K value is fixed at 0.

5.6 Polyatomic Molecular Vibrations

In order to account for degrees of freedom, without the center of mass translation, we have $3N-3$ degrees of freedom. Of these, 3 degrees of freedom are taken by the Euler angles to describe rotations, so the remaining $3N-6$ degrees of freedom must go into vibrational modes of the molecule. We can define a coordinate system whose origin is at the center of mass of the molecule, and atoms are connected by a framework of Hooke's law forces, with small displacements for equilibrium. We can then generalize

the expression for potential energy as a sum over all $3N$ displacements.

$$V = V(0) + \sum_i \left(\frac{\partial V}{\partial x_i}\right)_0 x_i + \frac{1}{2}\sum_{i,j}\left(\frac{\partial^2 V}{\partial x_i \partial x_j}\right)_0 x_i x_j + \ldots \tag{5.41}$$

For small displacements, we can define $V(0) = 0$. In addition, the first derivative terms will be 0 at the minimum of potential, and we can define

$$V = \frac{1}{2}\sum_{i,j} k_{i,j} x_i x_j \tag{5.42}$$

$$k_{ij} = \left(\frac{\partial^2 V}{\partial x_i \partial x_j}\right)_0 \tag{5.43}$$

where k_{ij} is the generalized force constant. In order to simplify this problem, we introduce mass-weighted coordinates $q_i = m_i^{1/2} x_i$, where $m_i^{1/2}$ is the mass of the atom being displaced by distance x_i. We can then write the potential energy as

$$V = \frac{1}{2}\sum_{i,j} K_{ij} q_i q_j \tag{5.44}$$

$$K_{ij} = \frac{k_{ij}}{(m_i m_j)^{1/2}} = \left(\frac{\partial^2 V}{\partial q_i \partial q_j}\right)_0 \tag{5.45}$$

This expression gives the total kinetic energy to be

$$T = \frac{1}{2}\sum_i m_i^2 \dot{x}_i^2 = \frac{1}{2}\sum_i^2 \dot{q}_i^2 \tag{5.46}$$

The classical total energy expression then becomes

$$E = \frac{1}{2}\sum_i \dot{q}_i^2 + \frac{1}{2}\sum_i K_{ij} q_i q_j \tag{5.47}$$

Complications arise when there are cross-terms in the potential energy function. We can use normal mode analysis to identify linear combinations Q_i of the coordinates that are free of cross-terms so that the total energy becomes

$$E = \frac{1}{2}\sum_i \dot{Q}_i^2 + \frac{1}{2}\sum_i \kappa_i Q_i^2 \tag{5.48}$$

When we consider the quantum mechanical description of a polyatomic molecule, the Hamiltonian becomes a sum of terms expressed as

$$H = \sum_i H_i \tag{5.49}$$

where

$$H_i = -\frac{1}{2}\hbar^2 \frac{\partial^2}{\partial Q_i^2} + \frac{1}{2}\kappa_i Q_i^2 \tag{5.50}$$

To simplify the analysis of this problem, we can observe that the Hamiltonian is a sum of terms, so the vibrational wavefunction becomes a product of the individual wavefunctions:

$$\psi = \psi_{\nu_1}(Q_1)\psi_{\nu_2}(Q_2)\ldots =_i \psi_{\nu_i}(Q_i) \tag{5.51}$$

Each of the terms in the total wavefunction is the solution of the Schodinger equation

$$-\frac{1}{2}\hbar^2 \frac{\partial^2 \psi(Q_i)}{\partial Q_i^2} + \frac{1}{2}\kappa_i Q_i^2 \psi(Q_i) = E\psi(Q_i) \tag{5.52}$$

We can solve this equation with harmonic oscillator wavefunctions whose energy levels are

$$E_{\nu_i} = \left(v_i + \frac{1}{2}\right)\hbar\omega_i \tag{5.53}$$

where

$$\omega_i = \kappa_i^{1/2} \tag{5.54}$$

$$v_i = 0, 1, 2, \ldots \tag{5.55}$$

The corresponding wavefunctions are

$$\psi_{v_i} = N_{v_i} H_{v_i}(y_i)\, e^{-y_i^2/2} \tag{5.56}$$

$$y_i = \left(\frac{\omega_i}{h_{bar}}\right)^{1/2} Q_i \tag{5.57}$$

Here, N_{v_i} is a normalization constant and H_{v_i} is a Hermite polynomial. Therefore, the total vibrational energy of the molecule is

$$E = \sum_i \left(\nu_i + \frac{1}{2}\right) \hbar\omega \tag{5.58}$$

5.7 Polyatomic Molecular Vibration-Rotation

In order to consider the normal-mode aspect of the development of the vibration-rotation Hamiltonian, let us assume harmonic vibrations such that a transformation can be defined from the $3N$ mass-adjusted Cartesian nuclear displacements, $\sqrt{M_i}\Delta\alpha_i$, to the $3N-6$ normal coordinates $Q_{s\sigma}$:

$$Q_{s\sigma} = \sum_{i=1}^{N} \sum_{\alpha} (l_{is\sigma}^{\alpha})^{-1} \left(\sqrt{M_i}\Delta\alpha_i\right) \tag{5.59}$$

or

$$\sqrt{M_i}\Delta\alpha_i = \sum_{s\sigma} l_{is\sigma}^{\alpha} Q_{s\sigma} \tag{5.60}$$

Here, the subscript s denotes the sth normal mode with frequency $\omega_s = \lambda_s^{1/2}/2\pi c$, and σ describes the degree of degeneracy of mode s. For example, if a mode is not degenerate, then $\sigma = 1$; if the mode is doubly degenerate, $\sigma = 2$; if the mode is triply degenerate, $\sigma = 3$.

The momentum conjugate to normal coordinate $Q_{s\sigma}$ is:

$$p_{s\sigma} = -i\hbar \frac{\partial}{\partial Q_{s\sigma}} \tag{5.61}$$

The components of internal angular momentum p associated with vibration are defined by

$$p = \sum_{s\sigma} \sum_{s'\sigma'} \zeta^{\alpha}_{s\sigma s'\sigma'} Q_{s\sigma} Q_{s'\sigma'} \tag{5.62}$$

Here, $\zeta^{\alpha}_{s\sigma s'\sigma'}$ is the Coriolis coupling constant that links vibrations $s\sigma$ to $s'\sigma'$. The coupling constant depends on the geometry of the molecule and is defined as:

$$\zeta^{\alpha}_{s\sigma s'\sigma'} = \sum_{i=1}^{N} \left(l^{\beta}_{is\sigma} l^{\gamma}_{is'\sigma'} - l^{\gamma}_{is\sigma} l^{\beta}_{is'\sigma'} \right) \tag{5.63}$$

In this equation, α, β, and γ are cyclic.

In our analysis, $\mathbf{P}$ is the total angular momentum and $\mathbf{p}$ is the angular momentum of vibration, so $\mathbf{P} - \mathbf{p}$ is the angular momentum of rotation. The rotation angular momentum related to the angular velocity is

$$\boldsymbol{\omega} = \boldsymbol{\mu} \cdot (\mathbf{P} - \mathbf{p}) \tag{5.64}$$

or

$$\omega_{\alpha} = \sum_{\beta} \mu_{\alpha\beta} (P_{\alpha} - p_{\alpha}) \tag{5.65}$$

Here, μ is the reciprocal of the inertia tensor. This formulation of the vibration-rotation energy can now be written as a Hamiltonian:

$$E = \frac{1}{2}\sum_{\alpha\beta}\mu_{\alpha\beta}(P_\alpha - p_\alpha)(P_\beta - p_\beta) + \frac{1}{2}\sum_{s\sigma} p_{s\sigma}^{*2} + V \tag{5.66}$$

The potential term V can be expanded as a Taylor series in normal coordinates $Q_{s\sigma}$:

$$V = \frac{1}{2}\sum_{s\sigma}\lambda_s Q_{s\sigma}^2 + \sum_{s\sigma}\sum_{s'\sigma'}\sum_{s''\sigma''} K_{s\sigma s'\sigma' s''\sigma''} Q_{s\sigma} Q_{s'\sigma'} Q_{s''\sigma''} + \ldots \tag{5.67}$$

It can be shown that directly substituting the quantum mechanical operators into the classical Hamiltonian leads to a Hamiltonian that is not Hermitian. Darling and Dennison showed that the quantum mechanical Hamiltonian can be expressed in a Hermitian form as

$$H = \frac{1}{2}\mu^{1/4}\sum_{\alpha\beta}(P_\alpha - p_\alpha)\mu^{-1/2}\ \mu_{\alpha\beta}\ (P_\beta - p_\beta)\mu^{-1/4} + \frac{1}{2}\mu^{1/4}\sum_{s\sigma} p_{s\sigma}^*\ \mu^{-1/2}\mu_{\alpha\beta}\ p_{s\sigma}^*\ \mu^{-1/4} + V \tag{5.68}$$

where $\mu = \det(\mu_{\alpha\beta})$. Later, Watson used the commutation relations to show that the Hamiltonian can be reduced to

$$H = \frac{1}{2}\sum_{\alpha\beta}(P_\alpha - p_\alpha)\mu_{\alpha\beta}(P_\beta - p_\beta) + \frac{1}{2}\sum_{s\sigma} p_{s\sigma}^{*2} + U + V \tag{5.69}$$

where U is a complicated arrangement of commutators that is a function only of coordinates but not momenta.

The resulting Schrodinger equation from both the Darling and Dennison (1940) form and the Watson form cannot be solved analytically, so approximations have to be made that can be compared with experimental results. The strategy is to expand the Hamiltonian in normal coordinates so that consecutive terms change by

approximately one order of magnitude. Then, contact transformations are applied in order to more easily obtain energies to orders higher than the first. This type of analysis was pioneered by Amat, Nielsen, and Goldsmith, and subsequently applied to molecules with 3-fold symmetry by Tarrago (Amat et al. 1971).

The lowest-order terms in the analysis should meet the following conditions:

1. The equation should be simple enough that it can be analyzed

2. The lowest-order results should explain overall features in experimental data

The approximation that regards the molecule as a rigid rotor plus a set of uncoupled harmonic oscillators satisfies these conditions. Furthermore, the rigid-rotor, harmonic oscillator representation can be expanded to higher-order terms by expressing $\mu_{\alpha\beta}$ in terms of normal coordinates:

$$\mu_{\alpha\beta} = \frac{1}{I_\alpha I_\beta}\left[\Omega^{(0)\alpha\beta} + \sum_{s\sigma}\Omega^{(1)\alpha\beta}_{s\sigma}Q_{s\sigma} + \sum_{s\sigma}\sum_{s'\sigma'}\Omega^{(2)\alpha\beta}_{s\sigma s'\sigma'}Q_{s\sigma}Q_{s'\sigma'} + \quad \ldots\right] \tag{5.70}$$

The $\Omega^{(n)\alpha\beta}_{s\sigma s'\sigma'\ldots}$ terms are complex functions of molecular parameters. We note, however, that the principal axis system is used for the inertia tensor, so

$$\Omega^{(0)\alpha\beta} = I_\alpha \delta_{\alpha\beta} \tag{5.71}$$

where I_α is the equilibrium moment of inertia about the αth axis. Hence,

$$\mu_{\alpha\beta} = \frac{1}{I_\alpha}\delta_{\alpha\beta} \tag{5.72}$$

We can now examine how molecular vibration-rotation theory can be translated into a form that is useful for least-squares regression analysis.

5.8 Analysis of Vibration-Rotation Spectra

While the proceeding formalism is correct, it is not necessarily practically useful for analyzing real molecular data. Following the strategy of Amat and Nielsen (1958, 1961, 1962), we can derive equations that describe vibration-rotation energy levels of a polyatomic molecule. We start with a Hamiltonian that is expanded in a power series .

$$H = h_0 + H_1 + H_2 + H_3 + \ldots \tag{5.73}$$

This Hamiltonian is diagonal with respect to quantum numbers J and M, but not necessarily diagonal with respect to ν_s, l_s, m_s, and K. We can then make a contact transformation, which results in a Hamiltonian expression that is diagonal with respect to all quantum numbers for an axially symmetric molecule.

$$H = H_0 + h_1^{'} + h_2^{'} + h_3^{'} + \ldots \tag{5.74}$$

We can then perform a second contact transformation, yielding

$$h^{+} = H_0 + h_1^{'} + h_2^{+} + h_3^{+} + \ldots \tag{5.75}$$

This Hamiltonian expression is diagonal with respect to all quantum numbers through h_1' for axially symmetric molecules. In the absence of accidental resonances, off diagonal terms in h_2^+ generally do not contribute to energy before the fourth order, and off diagonal terms in h_3^+ do not contribute to energy before the sixth order.

From the transformed Hamiltonian expression, we can derive equations that calculate all vibration-rotation transitions for a symmetric top molecule. The analyst can then use this generalized transition frequency expression to subject observed unperturbed spectral data to least-squares analysis methods. Moreover, in the presence

of perturbations, any subset of unperturbed lines can be analyzed in the same way using the simultaneous transition frequency expressions.

If the energy levels of an axially-symmetric molecule are denoted by quantum numbers

$$\{\nu_n, \nu_{n+1}, \ldots, \nu_t, l_t, \ldots, J, K, M\}$$

then the transition

$$\begin{aligned} & \{\nu_n + \Delta\nu_n, \ldots, \nu_t + \Delta\nu_t, l_t + \Delta l_t, \ldots, J + \Delta J, K + \Delta K, M + \Delta M\} \\ \Leftarrow \quad & \{\nu_n, \ldots, \nu_t, l_t, \ldots, J, K, M\} \end{aligned}$$

is a change from the lower state to the upper state with a corresponding energy change energy change $E'_{VR} \Rightarrow E''_{VR}$. The transition frequency expression will be

$$\begin{aligned} & (\nu_n, \nu_{n+1}, \ldots, \nu_t, l_t, \ldots, \Delta\nu_n, \ldots, \Delta\nu_t, \Delta l_t, \ldots, \Delta K, \Delta J, K, J) \\ = \quad & (\nu_n, \nu_{n+1}, \ldots, \nu_t, l_t, \ldots, \Delta\nu_n, \ldots, \Delta\nu_t, \Delta l_t, \ldots)\, \Delta K_{\Delta J_K}(J) \\ = \quad & E'_{VR} - E''_{VR} \end{aligned}$$

The generalized transition frequency is derived from subtracting the differences in energy. Expressions for specific cases of vibrational bands are given in the tables at the end Blass' 1976 paper. The major constants found in the energy expression are given as follows.

$$B_0 = B'' = B_e - \sum_s \frac{g_s}{2}\alpha_s^B + \sum_{s,s}^{s \leq s'} \frac{g_{s'} g_s}{4}\gamma_{ss'}^B + \Delta B_e \tag{5.76}$$

$$B_\nu = B' = B_0 - \sum_s \nu_s \alpha_s^B + \sum_{s,s'}^{s \leq s'} \left(\nu_s \nu_{s'} + \frac{\nu_s g_{s'}}{2} + \frac{\nu_{s'} g_s}{2} \right) \gamma_{ss'}^B + \sum_{t,t'}^{t \leq t'} \gamma_{l_t l_{t'} l_t l_{t'}}^B \tag{5.77}$$

$$A_0 = A'' = A_e - \sum_s \frac{g_s}{2} \alpha_s^A + \sum_{s,s}^{s \leq s'} \frac{g_{s'} g_s}{4} \gamma_{ss'}^A + \Delta A_e \tag{5.78}$$

$$A_\nu = A' = A_0 - \sum_s \nu_s \alpha_s^A + \sum_{s,s'}^{s \leq s'} \left(\nu_s \nu_{s'} + \frac{\nu_s g_{s'}}{2} + \frac{\nu_{s'} g_s}{2} \right) \gamma_{ss'}^A + \sum_{t,t'}^{t \leq t'} \gamma_{l_t l_{t'} l_t l_{t'}}^A \tag{5.79}$$

$$D_0^m = D_m'' = D_e^m - \sum_s \frac{g_s}{2} \beta_s^m \tag{5.80}$$

$$D_\nu^m = D_m' = D_0^m - \sum_s \nu_s \beta_s^m \tag{5.81}$$

In order to use the simultaneous frequency expressions, we can regard quantum numbers $\{\nu_n, \nu_{n+1}, \ldots, \nu_t, l_t, \ldots, \Delta\nu_n, \ldots, \Delta\nu_t, \Delta l_t, \ldots, \Delta K, \Delta J, K, J\}$ as regressor variables in a linear regression equation. The transition frequency then becomes the response variable in this analysis scheme.

5.9 Other Considerations

In addition to analyzing the transition frequencies in spectral data, the researcher may wish to account for other observed effects. In order for a molecule to absorb or emit electromagnetic radiation, at least one component of

$$\mu = \langle \psi' \mid \mu_i \mid \psi'' \rangle \tag{5.82}$$

must be nonzero, where μ_i is the electric dipole moment. Symmetry properties of zero-order wavefunctions lead to the following selection rules. For parallel bands:

$$\Delta K = 0,\ \Delta J = 0, \pm 1 \tag{5.83}$$

if $K \neq 0$ and

$$\Delta J = \pm 1 \tag{5.84}$$

if $K = 0$. For perpendicular bands

$$\Delta K = \pm 1,\ \Delta J = 0, \pm 1 \tag{5.85}$$

When effects of higher-order terms take effect, a more complete set of selection can be derived. Following Amat et al. (1971), a symmetry-adapted set of wavefunctions become

$$\begin{aligned} \mid \ + \ \rangle &= \mid \{\nu_s\}, \{l_s\}, K, J \ \rangle \ + \ \mid \{\nu_s\}, \{-l_s\}, -K, J \ \rangle \\ \mid \ - \ \rangle &= \mid \{\nu_s\}, \{l_s\}, K, J \ \rangle \ - \ \mid \{\nu_s\}, \{-l_s\}, -K, J \ \rangle \end{aligned}$$

The intensities of spectral lines are can be calculated from transition moments. If $F(K, J, \Delta K, \Delta J)$ defines the transition moment, then the intensity of the line becomes

$$\nu G F(K, J, \Delta K, \Delta J) \exp\left[-\frac{E_0(J, K)}{kT}\right] \tag{5.86}$$

where ν is the transition frequency, G is a statistical factor that depends on the spin of the nuclei and the degeneracy of the ground state, $E_0(J, K)$ is the ground state energy, k is the Boltzman constant and T is the absolute temperature.

Previously, transition moment integrals have been calculated. The results can be given as follows. For the case that $\Delta K = \pm 1$, If $\Delta J = 1$:

$$F(K, J, \Delta K, \Delta J) = \frac{(J + 2 + K\Delta K)(J + 1 + K\Delta K)}{(J + 1)(2J + 1)} \tag{5.87}$$

If $\Delta J = 0$:

$$F(K, J, \Delta K, \Delta J) = \frac{(J + 2 + K\Delta K)(J + K\Delta K)}{J(J + 1)} \tag{5.88}$$

If $\Delta J = -1$:

$$F(K, J, \Delta K, \Delta J) = \frac{(J - 1 - K\Delta K)(J - K\Delta K)}{2(2J + 1)} \tag{5.89}$$

While for the case that $\Delta K = 0$, If $\Delta J = 1$:

$$F(K, J, 0, \Delta J) = \frac{(J + 1)^2 - K^2}{(J + 1)(2J + 1)} \tag{5.90}$$

If $\Delta J = 0$:

$$F(K, J, 0, \Delta J) = \frac{K^2}{J(J + 1)} \tag{5.91}$$

If $\Delta J = -1$:

$$F(K, J, 0, \Delta J) = \frac{J^2 - K^2}{2J\,(2J+1)} \tag{5.92}$$

The proceeding equations allow us to determine the systematic structure of vibration-rotation bands. Historical convention denotes ΔK and ΔJ transitions of -1,0, and 1 as P,Q, and R, respectively. Transitions are symbolized in the form ${}^{\Delta K}\Delta J_K\,(J)$. Hence, the label ${}^{R}Q_K\,(J)$ signifies a transition from the K, J level of the ground vibrational state to the $K+1$, J level of an excited vibrational state. The consequence of the intensity equations is that, for a fixed K value, the intensity of the spectral lines is the product of a linearly increasing function of J with an exponentially decreasing function of J (Blass and Nielsen 1974). This causes the intensity to grow with increasing J, reach a maximum, and then decrease. The maximum progressively approaches $J = K$ as K increases.

The positions of lines are also affected by sequential K and J values. The separation of adjacent lines in a K-subband is

$$\begin{aligned} \Delta\nu \;=\; & 2\left[B_0\Delta J - \sum \alpha_s\nu_s\,(J+\Delta J+1)\right] - \\ & 4D_0^J\left[(J+\Delta J+1)^3 - (J+1)^3\right] - \\ & 2D_0^{JK}\left[(K+\Delta K)^2\,(J+\Delta J) - K^2\,(J+1)\right] \end{aligned} \tag{5.93}$$

This equation implies that the separation between adjacent lines varies approximately linearly with J (Blass and Nielsen 1974). These properties, combined with intensity structures, can be aids in the assignment of subbands as researchers try to match observed transition lines to quantum numbers.

5.10 Conclusion

This chapter developed the equations that describe vibration-rotation spectra. It then showed how these Hamiltonian expressions can be put into forms that are practically useful for data analysis. In order to use the transition expressions in regression analysis, they can be put into a form that minimizes linear dependencies (Blass 1976). Blass' 1976 paper lists a catalog of equations for specific cases of axially-symmetric molecules where transition expressions can be used for regression analysis. The interested reader can consult this paper for more information about the relationship between molecular theory and regression analysis.

Chapter 6

Regression of Power Series

6.1 Introduction

Many physical processes are modeled as Taylor or power series expansions. Oscillatory motion in classical and quantum physics often are expanded Taylor series around the minimum of some potential function. While this power series approach can be useful for developing theoretical models from first principles, how can researchers use observed data to subject these derived models to statistical analysis? Variable selection in multivariate regression offers one way of testing which variables are most important in describing some physical process.

Variable selection in classical multivariate regression has been popular since the 1960's. Efroymson (1960) proposed one of the first stepwise regression analysis procedures, where variables in a regression model are iteratively added and tested for their significance. Under the classical paradigm, variable significance is tested by the reduction in sum of squared residual error. Blass (1963) and Boyd (1963) implemented Efroymson's procedure to test a power series expansion in molecular models. Kurlat (1969) revised this analysis system and applied it to axially symmetric molecules.

This system was also used by Kurlat et al. (1971) to analyze CD_3I. Hafford (1972) implemented this stepwise regression analysis system with iteratively recomputed weights.

Although implementing the multivariate regression analysis as a forward stepwise process was a great advance, by 1970, some criticisms appears in the literature. Mantel (1970) remarked that the order of the stepwise process generally affects the final model. Others (Boyce et al. 1974, Wilkinson 1989, pg 177-178) criticized the arbitrary levels of significance for variables entering or leaving the model. The stepwise process is an ad hoc analysis system with little basis in statistical theory. Changing the order in which the variables enter the model equation or changing the thresholds for variable entry or exit can cause the algorithm to compute radically different final models for the same data set. In addition, because classical regression methods try to minimize squared residual error, it generally overfits data, computing models that generalize poorly.

In order to address these issues, Akaike (1973) introduced a new way of judging models computed in the regression process. Instead of relying on levels of significance, different combinations of model parameters are assigned a score. The scoring function, called Akaike Information Criterium (AIC) is expressed as

$$AIC(m) = -2\log(\theta) + 2m \tag{6.1}$$

where the first term is double the maximized log-likelihood estimate of the regression and the second term is twice the number of free parameters. AIC is a penalized cost function. The two terms counteract each other, trying to find the optimal balance between lack of fit and lack of parsimony in the regression analysis. Later, Bozdogan (1988, 1990a, 2000, 2004) proposed a different scoring function called $ICOMP$, which

is given as

$$ICOMP = -2logL(\Theta_k) + 2C(\Sigma_{Model}) \tag{6.2}$$

The first term in $ICOMP$ accounts for lack of fit in the same way as the first term in AIC does. The second term, however, measures the complexity of terms included in the regression model. While AIC only penalizes twice the number of included parameters, $ICOMP$ penalizes for interactions between model parameters in the regression model. $ICOMP$ is currently considered by specialists to be the most correct scoring function in statistical model selection (Bozdogan, private communication)

We propose using these scoring functions to find the best model in a power-series modeling scenario. Instead of using the F test based stepwise procedure, we will assign combinations of model terms complexity scores. The model combination that achieves the minimum score is the best for describing the system under study. Section 2 of this chapter will review stepwise analysis of power series while section 3 will introduce the information scoring method of analyzing power series. Section 4 will show how we can implement a Beaton-Tukey (1974) weighting technique in information scored regression. The chapter will conclude in section 5. The next chapter will highlight examples of data sets processed with this new algorithm.

6.2 Stepwise Regression and Power Series

Regression analysis of power series differs from the general problem of regression because of the different orders of magnitude of the variables. Although in general regression studies, every variable combination is a potential model, the power series regression requires that lower order terms be in the model if higher order terms are in it. Beaton and Tukey (1974) give a good description of this problem in their paper. As an example, they describe the process of analyzing the spectrum of diatomic

molecules used by Mann et al. (1961). In this process, the vibrational energies E of the molecule are given by

$$\begin{aligned} E &= \nu_0 + B(J(J+1) - D(J(J+1))^2 + H(J(J+1))^3 \\ &\quad + I_1(J(J+1))^4 + I_2(J(J+1))^5 + I_3(J(J+1))^6 + \ldots \end{aligned} \tag{6.3}$$

where J is the rotation quantum number of the molecule and the coefficients are the molecular constants. The researcher must use a regression analysis procedure to calculate values for these molecular constants. Depending on the complexity of the molecule, there may be many terms in the respective orders of magnitude, and some variable selection method is required to give the best equation that describes the observed molecular spectrum.

Stepwise regression is a classical procedure for selecting variables in a regression equation. Stepwise regression iteratively adds or deletes variables to a model and tests the resulting reduction in sum of squared error. In order to use stepwise regression, variable entry and exit thresholds must be defined. There are several versions of stepwise regression procedures. Forward stepwise regression successively adds variables to an equation while backwards stepwise regression successively deletes them. In mixed stepwise regression, the user starts with a preset number of variables and the algorithm iteratively adds or deletes them until no change occurs. In all of the stepwise procedures, the candidate variable's F ratio for entry is computed. If the F ratio is above the threshold, the variable enters, otherwise, it does not. Likewise, for variable deletion, a similar F test is performed. If the F value is below the threshold, it is deleted, otherwise, it remains in the regression equation.

Efroymson (1960) originally proposed a forward procedure that successively evaluates the F ratios of each variable. Blass (1963) and Boyd (1963) were among the first to implement a modified version of this procedure for analysis of quantum mechanical power series expressions. Their regression analysis examined contributions of terms in the power series expansion. The independent variables were functions of molecular quantum numbers and the dependent variable was the observed transition frequency. Kurlat et al. (1971) used a similar approach in their analysis system. The implementation of their algorithm is as follows:

1. Insert variables that the researcher believes are important. These may be the lowest order terms for which the researcher already has good parameter estimations. These terms are forced to be in the model at each iteration of the algorithm.

2. Find the variable that is not currently in the model that has the largest F statistic value to enter. The stepwise algorithm used by Blass and others sequentially adds variables based on their order in the power series expression. If there are no variables in the model that have an F value as large as the F_{in} value, then stop.

3. Find the variable in the model, other than those forced to be in, that has the smallest F statistic value to remove. If this value is less than F_{out}, then remove this variable from the model. Repeat this procedure until no more variables are dropped, then go to step 2.

Because these procedures analyze power series expansions, generally only the highest order terms need to be tested for significance. The modified forward stepwise strategy was that the researcher could force lower order terms to be in the model and successively add higher order terms. We propose implementing a structure that uses

information scoring and includes lower order terms but tests combinations of high order terms. The next section describes this new approach.

6.3 Information Scoring in Power Series

Since its introduction, researchers have enjoyed the benefits of using information scores in regression studies. Instead of defining a hypothesis based test statistic, under the information based regression paradigm, the analysis algorithm assigns scores that judge the quality of the regression model. AIC (Akaike 1973) was the first example of a regression scoring function, and it is expressed as

$$AIC(m) = -2\log(\theta) + 2m \tag{6.4}$$

For n data points having normally and independently distributed errors with residual variance $\hat{\sigma}^2$, AIC is given as

$$AIC(m) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2(m+1) \tag{6.5}$$

ICOMP is another scoring function that is used in statistical model selection. The ICOMP scoring function is given as

$$ICOMP = -2\log(\theta) + 2C(F^{-1}) \tag{6.6}$$

Likewise, under the same residual assumptions, we have

$$ICOMP(REG) = n\log(2\pi) + n\log(\hat{\sigma}^2) + n + 2C_1((\mathbf{X}'\mathbf{X})^{-1}) \tag{6.7}$$

Here, the complexity operation is defined as

$$C_1((\mathbf{X}'\mathbf{X})^{-1}) = \frac{q}{2}\log\left[\frac{tr((\mathbf{X}'\mathbf{X})^{-1})}{q}\right] - \frac{1}{2}\log\left[\det((\mathbf{X}'\mathbf{X})^{-1})\right] \quad (6.8)$$

with $q = rank((\mathbf{X}'\mathbf{X})^{-1})$.

In order to use these scoring functions, we must compute the scores for different combinations of regression equations. However, unlike general regression studies where researchers consider every possible combination of model parameters, we only want to consider combinations of model parameters that obey power series expansions. If the researcher knows that some low order terms must be included in the model, but wants to test the contributions of higher order terms, then under the information scoring paradigm, we want to assign scores to these combinations of model parameters.

Under the general information based regression method, subsetting generates every possible combination of model parameters. There are $2^k - 1$ possible combinations of model equations for k variables. However, in a power series analysis situation, we always include low order terms. We can therefore devise a scheme that only tests combinations of higher-order terms. Under our power series subsetting scheme, we always include terms 1 through m and test all combinations of model parameters $m + 1$ through k. This approach applies the discrimination ability of information scoring to the special considerations of modeling under a power series expansion. This process is summarized as follows:

1. Generate a sequence of regression model equations that include lower order terms 1 through m but has every combination of higher order variables $m + 1$ through k.

2. Assign each of these combinations an information score with one of the defined scoring functions. Under normal regression modeling assumptions, ICOMP is considered to be the best choice.

3. The model combination with the minimum ICOMP score is the best to describe the system.

Although the subsetting paradigm works well in finding the minimum information scores, the number of models that need to be scored grows quickly with k. Even for a moderate value of k, the number of possible model calculations that must be evaluated quickly becomes impractically large for most computers. As an alternative to subsetting, Bearse and Bozdogan (2002) proposed implementing information scored regression as a binary genetic algorithm (GA). Under this method, binary strings represent possible combinations of model parameters, with 1 denoting inclusion and 0 denoting exclusion. The populations of strings undergoes standard genetic operations of mutation and recombination, and the scoring function $ICOMP$ acts as the fitness function that ranks the quality of possible model combinations. The probability of reproduction varies with fitness. Bearse and Bozdogan demonstrated that their method could quickly find optimal models for regression cases with over 100 model variables, a task that currently is virtually impossible under the standard scored subsetting method (Bearse and Bozdogan 2002).

We propose that the power series regression can be implemented in a GA framework in an analgous way as the power series subsetting method. Our proposed method constructs a model where low order terms are forced to be in the model, but higher order terms are represented in the GA method. This applies the efficient searching capability of the GA to the analysis of power series. A description of our method follows:

1. Generate a population of binary strings that include lower order terms 1 through m but has different combinations of higher order variables $m+1$ through k.

2. Assign each of these combinations an information score with one of the defined scoring functions. Rank the members of this population according to fitness by an information score. ICOMP is generally regarded to be the best scoring function under standard modeling assumptions.

3. Apply the GA operations of mutation, recombination, and reproduction to this population. Continue this loop until convergence of the GA population.

4. The model combination that converges to the minimum ICOMP score is the best to describe the system

6.4 Weighting in Scored Regression

Beaton and Tukey (1974) described a method of modeling perturbed spectra with weighted regression. While the power series expression (6.3) describes the predicted energies of molecular transitions, this type of power series expansion ignores interactions between molecular states. When perturbations are not a serious problem, the transition frequencies predicted by the power series model follow a regression line well. Perturbed points, however, can cause transition energies to fall far away from the regression line. In order to reduce the effects of these perturbations, Beaton and Tukey defined the least squares regression estimates to be

$$\hat{\boldsymbol{\beta}} = (\mathbf{X'WX})^{-1}\mathbf{X'Wy} \tag{6.9}$$

here, the weight matrix $\mathbf{W}$ is a diagonal matrix whose elements are iteratively recalculated. Let ε_i be the standardized residual value of the ith data point. Under

the Beaton-Tukey method, a limit $s_{\max}$ of residuals from the regression line defines the maximum distance from the regression line. The weights are then recalculated at each iteration.

$\mathbf{W}_{ii} = \left(\frac{1}{(1-\varepsilon_i)^2}\right)^2$ for $|\varepsilon_i| < s_{\max}$

$\mathbf{W}_{ii} = 0$ for $|\varepsilon_i| > s_{\max}$

We implemented this type of weighting scheme in our scored information regression. However, we defined two limits $s_{\min}$ and $s_{\max}$ in our method, so that our weights are updated as:

$\mathbf{W}_{ii} = 1$ for $|\varepsilon_i| < s_{\min}$

$\mathbf{W}_{ii} = \left(\frac{1}{(1-\varepsilon_i)^2}\right)^2$ for $s_{\min} < |\varepsilon_i| < s_{\max}$

$\mathbf{W}_{ii} = 0$ for $|\varepsilon_i| > s_{\max}$

These weights are updated for the best model at each iteration of the algorithm. The points with residual values greater than $s_{\max}$ are considered to be outliers. The scored subset analysis algorithm is given by:

1. Generate a sequence of regression model equations that include lower order terms 1 through m but has every combination of higher order variables $m+1$ through k. Initialize all weight values to 1

2. Assign each of these combinations an information score with one of the defined scoring functions. Update the weight values according to the model with the best information score. Continue the process of reweighting and finding the best information score until there is no change in the outlier assignments.

3. The model combination with the minimum ICOMP score is the best to describe the system.

Similarly, the scored GA regression analysis algorithm is given by

1. Generate a population of binary strings that includes lower order terms 1 through m but has different combinations of higher order variables $m+1$ through k. Initialize all weight values to 1.

2. Assign each of these combinations an information score with one of the defined scoring functions. Rank the members of this population according to fitness by an information score. Update the weight values according to the model with the best information score.

3. Apply the GA operations of mutation, recombination, and reproduction to this population. Continue the process of reweighting and finding the best information score until there is no change in the outlier assignments.

4. The model combination that converges under the GA process to the minimum ICOMP score is the best to describe the system

6.5 Conclusion

This chapter reviewed power series regression based on classical regression, and also showed how we can cast power series regression into a modern information scoring framework. The interested reader can consult Beaton and Tukey's 1974 paper for more information about weighted regression and Bozdogan's 2004 text for more information about information scoring in regression.

Chapter 7

Scored Regression in Spectroscopy

7.1 Introduction

For the past four decades, spectroscopists have relied on regression analysis methodology to interpret and analyze molecular spectra (Blass 1963, Boyd 1963). Initially, graph paper and hand calculators were used in the numerical analysis. As the resolution and quantity of data increased, digital computers were employed to speed the process of spectral data interpretation. The computerized method of spectral analysis used stepwise multivariate linear regression to perform the analysis (M. Kurlat 1969, H. Kurlat 1970, Hafford 1972). This method attempts to find the best terms to include in a molecular Hamiltonian equation by casting the equations in a form where the terms of the equation are linearly independent and can be analyzed by linear regression methodology (Blass 1976). This method was used effectively through the 1960's and 1970's. The terms in the molecular Hamiltonian arose form a power series expansion of the twice-transformed Hamiltonian (Blass and Nielsen 1974). During the 1960's and 70's, the resolution of the data was sufficient that Hamiltonian terms

through second order were resolvable. The observed effects led to molecular Hamiltonian equations with up to approximately 15 terms that adequately described the molecules under study. These terms included the main effects of the band origin and the second order distortion constants. The data sets generally had hundreds of identifiable transitions, and researchers analyzed the spectroscopic data using least-squares analysis methods to calculate molecular constants appropriate to multiple bands simultaneously.

This chapter will compare the historical stepwise results with analysis based on modern regression methodology. For relatively low resolution data, the regression analysis is sufficient for calculating all spectral parameters. For more modern, higher resolution data, the regression analysis would be the initial step in the analysis. The researcher would then iteratively analyze perturbations and recalculate molecular parameters.

7.2 Boyd/Kurlat CD_3I $2\nu_4$ Spectrum

The first data set that we compared was the $2\nu_4$ spectral band of CD_3I. This data was was originally analyzed by Boyd (1963) in his Ph.D. dissertation, and then reanalyzed by Kurlat et al. (1971). This set was acquired from a Littrow grating spectrometer at Michigan State University in the 1960's. These data had a resolution of $\sim$0.06 cm^{-1} and were calibrated against argon emission lines in the third and fourth order. When the data were recorded, the pressure was $\sim$40 Torr and the absorbing path length was approximately 8m (Kurlat et al. 1971). This data set was ideal for a first test case because at this resolution, it is almost unperturbed and is a textbook example of a statistical analysis of a molecular spectrum. This data set had 310 transitions. We used the same variable definitions as Kurlat et al., and applied the

Table 7.1: Definition of Molecular Parameters for $2\nu_4$

Variable	Parameter	Quantum Dependency
1	$\nu_4 + x_{44} + x_{l_4 l_4}$	$\Delta\nu_4$
2	$A_0 + 2A_e\zeta_4$	$(2K + \Delta K)\,\Delta K$
3	D_0^K	$K^4 - (K + \Delta K)^4$
4	α_4^A	$-\Delta\nu_4\,(K + \Delta K)^2$
5	α_4^B	$-\Delta\nu_4[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2]$
6	$x_{l_4 l_4} + (1/4)A_e\zeta_4$	Δl_4^2

subset analysis method to this data. We used the same values of B_0, D_0^J, and D_0^{JK} as the original authors. These values are set by microwave spectral studies which return much more precise parameter estimates. The values quoted by the original authors are $B_0 = 0.201482$ cm^{-1}, $D_0^J = 1.19 \times 10^{-7}$cm^{-1} and $D_0^{JK} = 1.612 \times 10^{-6}$cm^{-1}. Because of the physical interpretation of molecular constants, we forced the first three terms to be in the model, and let the algorithm select the remaining terms. Table 7.1 shows the respective molecular parameters in this analysis. Note that there is a sign correction in the α_4^B terms as compared with the Kurlat et al. paper.

The result of the subset analysis gave almost the same results as the Kurlat et al. analysis, both in numerical parameter values and in variables selected as meaningful in the regression. Table 7.2 gives the results from the unweighted regression. The numerical values have units of cm^{-1} and the confidence intervals are simultaneous confidence regions for all the given parameters. Table 7.3 shows the results of the weighted regression using our modified Beaton-Tukey scheme. For both calculations, we used 3.0 standard deviations as the outlier deletion limit, and in the weighted case, we used 1.0 standard deviations as the inner weight limit. The residuals for both regression cases showed no unusual features and had nearly ideal normal quantile plots.

Table 7.2: Unweighted Regression Estimates of Molecular Parameters of $2\nu_4$

Variable	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	2273.069	0.001	2273.070	0.001
2	3.4723	0.0002	3.4721	0.0003
3	3.81×10^{-5}	5×10^{-7}	3.77×10^{-5}	9×10^{-7}
4	0.01283	1×10^{-5}	0.01284	2×10^{-5}
5	8.7×10^{-5}	2×10^{-6}	8.7×10^{-5}	1×10^{-6}
6	8.8136	8×10^{-4}	8.8140	1×10^{-3}

Table 7.3: Weighted Regression Estimates of Molecular Parameters of $2\nu_4$

Variable	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	2273.069	0.001	2273.069	0.001
2	3.4723	0.0002	3.4720	0.0003
3	3.81×10^{-5}	5×10^{-7}	3.75×10^{-5}	9×10^{-7}
4	0.01283	1×10^{-5}	0.01284	2×10^{-5}
5	8.7×10^{-5}	2×10^{-6}	8.7×10^{-5}	1×10^{-6}
6	8.8136	8×10^{-4}	8.8144	1×10^{-3}

We can see that the information based subset analysis gave almost the same parameter estimates as the sum of squared error based analysis on this data set. The modern analysis gave a mean squared error value of $1.1\times10^{-4}\text{cm}^{-1}$. The results of this case confirm that in the low variable limit, the classical stepwise regression and the modern scored regression return nearly the same results.

7.3 Kurlat CD_3I $\nu_4 + \nu_5$ Spectrum

The next data set that provides a good test case is the $\nu_4 + \nu_5$ combination band of Methyl Iodide, measured from the Michigan State Littrow spectrometer. The data set had 389 transitions. We again subjected this data set to the information based

Table 7.4: Definition of Molecular Parameters for $\nu_4 + \nu_5$

Variable	Parameter	Quantum Dependency
1	$\nu_4 + \nu_5 + A_e(\zeta_4 + \zeta_4) + ...$	$\Delta\nu_4$
2	$A_0 + A_e(\zeta_4 + \zeta_4)$	$(2K + \Delta K)\,\Delta K$
3	D_0^K	$K^4 - (K + \Delta K)^4$
4	$\alpha_4^A + \alpha_5^A$	$-\Delta\nu_4\,(K + \Delta K)^2$
5	$\alpha_4^B + \alpha_5^5$	$-\Delta\nu_4[(J + \Delta J)(J + \Delta J + 1) - (K + \Delta K)^2]$

subset analysis. The results of the analysis follow. The numerical values have units of cm^{-1} and the confidence intervals are simultaneous confidence regions for all the given parameters. We note a sign correction in variable 5 that differs from the Kurlat et al. paper. The molecular parameter definitions are given in table 7.4. We again included the first three terms in the model, and let the algorithm select the remaining terms.

We can see that both the weighted and unweighted regression analysis returned values that were very close to the values calculated by the original authors. The mean squared error of both regression cases was 0.0015. Tables 7.5 and 7.6 show the results from the respective analysis trials. The reason for this difference comes from the final points identified as outliers. This data set showed perturbations, so we used the outlier limit of 2.0 standard deviations for both the weighted and unweighted analysis, and 1.0 standard deviations for the inner weight limit of the weighted trial. The residual plots did not show any unusual patterns and the residual quantile plots showed no problems with normality.

These results again verify that the information based analysis give almost the same results as the sum of squared error based analysis method. The same variables were selected using the different methods and the numerical parameter values agree well.

Table 7.5: Unweighted Regression Estimates of Molecular Parameters for $\nu_4 + \nu_5$

Variable	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	3337.494	0.005	3337.494	0.005
2	2.2097	0.0004	2.2095	0.0005
3	3.4×10^{-5}	1.5×10^{-6}	3.4×10^{-5}	2.5×10^{-6}
4	0.02818	8×10^{-5}	0.02821	1×10^{-4}
5	-3.6×10^{-4}	2×10^{-5}	-3.6×10^{-4}	1×10^{-5}

Table 7.6: Weighted Regression Estimates of Molecular Parameters for $\nu_4 + \nu_5$

Variable	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	3337.494	0.005	3337.499	0.005
2	2.2097	0.0004	2.2095	0.0005
3	3.4×10^{-5}	1.5×10^{-6}	3.4×10^{-5}	2.5×10^{-6}
4	0.02818	8×10^{-5}	0.02825	1×10^{-4}
5	-3.6×10^{-4}	2×10^{-5}	-3.5×10^{-4}	1×10^{-5}

7.4 Kurlat CD_3I $2\nu_4$, $\nu_4 + \nu_5$ and $\nu_2 + \nu_4$ Spectrum

For the next example, we examined the simultaneous analysis of the combination band of the Methyl Iodide data and compared the information scored analysis with the results from the Kurlat et al. paper. These data contain the data from the $2\nu_4$ and $\nu_4 + \nu_5$ bands plus the $\nu_2 + \nu_4$ combination band, for a total of 940 transitions. Table 7.7 shows the first and second order molecular parameters with their quantum dependencies. To save space, only the leading term is given in the table. We note sign corrections identified in the Kurlat et al. paper.

Tables 7.8 and 7.9 show the weighted and unweighted information scored analysis through second order. In this case, we forced the first order terms and the D_0^K in the model, and let the subsetting algorithm select the second order terms. The

Table 7.7: Definition of Molecular Parameters for Combination Band to Second Order

Variable	Parameter	Quantum Dependency
1	$\nu_2 + ...$	$\Delta\nu_2$
2	$\nu_4 + ...$	$\Delta\nu_4$
3	$\nu_5 + ...$	$\Delta\nu_5$
4	A_0	$(2K + \Delta K)\,\Delta K$
5	$A_e\zeta_4^z$	$-2\Delta l_4\,(K + \Delta K)$
6	$A_e\zeta_5^z$	$-2\Delta l_5\,(K + \Delta K)$
7	D_0^K	$K^4 - (K + \Delta K)^4$
8	α_2^A	$-\Delta\nu_2\,(K + \Delta K)^2$
9	α_4^A	$-\Delta\nu_4\,(K + \Delta K)^2$
10	α_5^A	$-\Delta\nu_5\,(K + \Delta K)^2$
11	α_2^B	$-\Delta\nu_2\left[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2\right]$
12	α_4^B	$-\Delta\nu_4\left[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2\right]$
13	α_5^B	$-\Delta\nu_5\left[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2\right]$
14	$x_{l_4 l_4}$	Δl_4^2

Table 7.8: Unweighted Regression Estimates of Molecular Parameters for Combination Band to Second Order

Number	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	961.797	0.009	961.792	0.006
2	2273.071	0.003	2273.070	0.003
3	1056.201	0.007	1056.204	0.006
4	2.5792	0.0003	2.5792	0.0004
5	0.4461	0.0005	0.4463	0.0002
6	-0.815	0.0005	-0.816	0.0003
7	0.000036	0.00001	0.000036	1×10^{-6}
8	-0.0042	0.0003	-0.0042	0.0002
9	0.01287	0.00003	0.01287	0.00004
10	0.0153	0.0001	0.0154	0.0001
11	0.00178	0.00003	0.00175	0.00002
12	0.000086	0.000004	0.000086	0.000002
13	-0.00044	0.00002	-0.00044	0.00001
14	8.592	0.002	8.592	0.001

Table 7.9: Weighted Regression Estimates of Molecular Parameters for Combination Band to Second Order

Number	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	961.797	0.009	961.790	0.006
2	2273.071	0.003	2273.070	0.003
3	1056.203	0.007	1056.203	0.005
4	2.5792	0.0003	2.5787	0.0004
5	0.4461	0.0005	0.4465	0.0002
6	-0.815	0.0005	-0.815	0.0003
7	0.000036	0.00001	0.000036	1×10^{-6}
8	-0.0042	0.0003	-0.0040	0.0002
9	0.01287	0.00003	0.01287	0.00004
10	0.0153	0.0001	0.0154	0.0001
11	0.00178	0.00003	0.00175	0.00002
12	0.000086	0.000004	0.000086	0.000002
13	-0.00044	0.00002	-0.00045	0.00001
14	8.592	0.002	8.592	0.001

numerical values have units of cm^{-1} and the confidence intervals are simultaneous confidence regions for all the given parameters. We used 2.5 standard deviations as the outlier limit in both cases, and 1.0 standard deviations for the inner weight limit in the weighted regression. The mean squared error of both regression calculations was 0.0004. The residual plots for both showed no unusual features, and the normal quantile plots showed that the residuals followed normal distributions well.

We can see that both the weighted and unweighted results choose all of the possible variables. Both regression procedures also calculated numerical values that were all statistically equal to the stepwise procedure.

Another trial using this data set was the regression analysis of these data with the model expanded the third order. The possible model terms are given in table 7.10. For this trial, we forced the first 7 variables to be in the model, and used an ICOMP scored subset selection to identify the other terms. The results of the unweighted analysis are in table 7.11. For this trial, we used 2.5 standard deviations

Table 7.10: Definition of Molecular Parameters for Combination Band to Third Order

Variable	Parameter	Quantum Dependency
1	$\nu_2 + ...$	$\Delta\nu_2$
2	$\nu_4 + ...$	$\Delta\nu_4$
3	$\nu_5 + ...$	$\Delta\nu_5$
4	A_0	$(2K + \Delta K)\,\Delta K$
5	$A_e\zeta_4^z$	$-2\Delta l_4\,(K + \Delta K)$
6	$A_e\zeta_5^z$	$-2\Delta l_5\,(K + \Delta K)$
7	D_0^K	$K^4 - (K + \Delta K)^4$
8	α_2^A	$-\Delta\nu_2\,(K + \Delta K)^2$
9	α_4^A	$-\Delta\nu_4\,(K + \Delta K)^2$
10	α_5^A	$-\Delta\nu_5\,(K + \Delta K)^2$
11	α_2^B	$-\Delta\nu_2\left[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2\right]$
12	α_4^B	$-\Delta\nu_4\left[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2\right]$
13	α_5^B	$-\Delta\nu_5\left[(J + \Delta J)\,(J + \Delta J + 1) - (K + \Delta K)^2\right]$
14	$x_{l_4l_4} + ...$	$\Delta l_4^2(J + \Delta J + 1) - (K + \Delta K)^2$
15	η_5^J	$-2\Delta l_5\,(K + \Delta K)\,(J + \Delta J)\,(J + \Delta J + 1)$
16	η_4^K	$\Delta l_4\,(K + \Delta K)^3$
17	η_5^K	$\Delta l_5\,(K + \Delta K)^3$
18	η_4^J	$\Delta l_4\,(K + \Delta K)\,(J + \Delta J)\,(J + \Delta J + 1)$

Table 7.11: Unweighted Regression Estimates of Molecular Parameters for Combination Band to Third Order

Number	Kurlat et al. Value	95% CI	Wicker Value	95% CI
1	961.799	0.009	961.793	0.006
2	2273.071	0.003	2273.0696	0.003
3	1056.202	0.007	1056.206	0.005
4	2.5797	0.0003	2.5791	0.0004
5	0.4461	0.0005	0.4464	0.0002
6	-0.816	0.0005	-0.818	0.0005
7	Not Sig		0.000037	1×10^{-6}
8	-0.0042	0.0003	-0.0040	0.0002
9	0.01287	0.00003	0.01285	0.00004
10	0.0153	0.0001	0.0154	0.0001
11	0.00178	0.00003	0.00175	0.00002
12	0.000086	0.000004	0.000086	0.000002
13	-0.00044	0.00002	-0.00042	0.00001
14	8.592	0.002	8.592	0.001
15	-4×10^{-6}	3×10^{-6}	-1×10^{-5}	2×10^{-6}
16	4×10^{-6}	2×10^{-6}	Not Sig	
17	-0.00001	0.00001	Not Sig	
18	Not Sig		Not Sig	

as the outlier limit. Our diagnostic plots showed no problems with normality and no strange patterns. The mean squared error of the unweighted regression analysis was 0.0004.

When we applied the genetic algorithm version of the analysis algorithm, we retrieved the same results as the subsetting algorithm. This is consistent with the proper application of information scored regression analysis.

7.5 Guelachvili et al. CD_3I ν_4 Spectrum

This data set represents a modern study with high spectral resolution from Guelachvili et al. (1984). The resolution of this data set was 5.4×10^{-3} cm^{-1} with over 2100 identified transitions. The original authors do not explain what possible terms are

Table 7.12: Unweighted Regression Estimates of Molecular Parameters for ν_4 Spectrum

Constant	Guelachvili et al. Value	Wicker Value
ν_0	2298.54431	2298.54369
A_4	2.5825779	2.5960037
B_4	0.20139595	0.201406187
$A\zeta_4$	0.463800	0.463952

in the model, but only state the final numerical results of their analysis. In order to reanalyze this data set, we derived a regression model using the appropriate expression from Blass' 1976 paper. The original authors state that in their initial fit, they used 740 lines that they regarded as mostly unperturbed. We applied our regression analysis program with outlier deletion with the minimum number of data points set to be 740. We constrained the ground state constants to be the same as the original authors: $A_0 = 2.59608$ cm^{-1}, $B_0 = 0.2014825$ cm^{-1}, $D_0^J = 0.1244 \times 10^{-7}$cm^{-1}, $D_0^{JK} = 1.611 \times 10^{-6}$cm^{-1} , and $D_0^K = 19.8 \times 10^{-6}$cm^{-1}. Our analysis represents the initial values in what would be an iterative perturbation analysis. We can compare our initial values to the authors final values. Tables 7.12 and 7.13 show the results of the unweighted and weighted regressions respectively. The original authors state that their initial fit had a standard deviation of 9×10^{-4}. The standard deviation of our weighted and unweighted fits were 8×10^{-4} and 6×10^{-4} respectively, a bit smaller than the original authors. Our method required no preselection of points, but rather automatically selected the points to be included in the initial fit.

7.6 Conclusion

This chapter compared the results of some historical data sets using stepwise regression against modern information scored regression. Our results indicated that the

Table 7.13: Weighted Regression Estimates of Molecular Parameters ν_4 Spectrum

Constant	Guelachvili et al. Value	Wicker Value
ν_0	2298.54431	2298.54397
A_4	2.5825779	2.5826756
B_4	0.20139595	0.201404898
$A\zeta_4$	0.463800	0.464011

data sets with relatively few variables gave practically identical results as the classical stepwise results, whereas the more complex cases with a larger number of possible variables showed some differences. This is consistent with the expected behavior of the classical verses modern analysis algorithms. We believe that this method can be used in more complex analysis cases and also in more general studies in physical science.

Chapter 8

Genetic Algorithms for Hyperellipsoidal Clustering

8.1 Introduction

In automated classification, we consider the problem of partitioning the data based only on observed values. The problem of automated clustering of data is important in many fields, such as multivariate image analysis (Mao and Jain 1996), astronomical survey data (Roeder and Wasserman 1997, Zhang and Zhao 2004) and geographic climatology (Hoffman et al. 2005). Clustering methods try to find some underlying structure in the data, relying on a measure of similarity to judge how similar data points are to one another. The K-means algorithm (Macqueen 1967) has been widely used in pattern recognition problems, with the user only defining the number of clusters K. Under the K-means scheme, seed values or initial cluster centroids are calculated, which are initial points chosen that may be representative of some feature or have some special qualities with respect to cluster membership. The algorithm then assigns data points to clusters based on their Euclidean distance from the seeds. After

the user specifies seed values, the identified cluster assignments and calculated cluster centroids iterate until the number of data points that change between iterations falls below a threshold. Many variants of the traditional K-means algorithm (Jain and Dubes 1989) try to minimize a metric with respect to some cluster property, such as the number of data points in the cluster or the within cluster variance.

We can formalize the description of partitioned cluster analysis as follows. We want to partition d dimensional data vectors into K groups. Let $\{x_i, i = 1, 2, \ldots n\}$ be a set of n data vectors, and let x_{ij} be the jth value of vector x_i. Let $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$. Then, if the ith pattern is in the kth cluster,

$$w_{ik} = 1 \tag{8.1}$$

else

$$w_{ik} = 0 \tag{8.2}$$

The matrix of these values, $\mathbf{W} = [w_{ij}]$ has the attributes

$$w_{ij} \in \{0, 1\} \tag{8.3}$$

and

$$\sum_{j=1}^{K} w_{ij} = 1 \tag{8.4}$$

The question of how best to choose centroid seed values and the choice of metric minimization has always plagued researchers using the K-means algorithm. Under the traditional paradigm, choosing different seed values generally yields different partitions of the same data set. In addition, the metric spaces of cluster cost functions are generally complex, multivariate and nonlinear, and under the traditional K-means paradigm, there is no guarantee that the iterative method optimizes the metric chosen.

Krishna and Murty (1999) addressed the issue of optimizing cluster calculations while simultaneously overcoming the issue of initial choice of cluster seeds by introducing a version of the K-means algorithm based on Genetic Algorithm (GA) strings. The user randomly initializes a population of strings, with the strings denoting the cluster assignments for each data point. For example, if we want to partition a data set with 100 observations into 5 clusters, each entry in the string $\{x_1, x_2, \ldots, x_{100}\}$ can assume integer values 1 to 5, and there is one string entry for each data point. The strings in the Genetic K-Means algorithm represent candidate solutions to the problem under study. Using a GA population, the strings undergo genetic-type operations, where values in the string undergo random changes and recombinations.

An extension of the K-Means method occurs in Hyperellipsoidal Clustering (HEC). In HEC, clusters generally have different variances in different directions. This is like having clusters with football or cigar shapes, which contrasts with the K-means clustering process, where clustering based on Euclidean distance leads to hyperspherical clusters (Wang et al. 1997, Wang and Xia 1997). One is handicapped when using K-Means in HEC because of the more complex covariance structure. In the mid 1990's, Mao and Jain (1996) proposed handling the HEC problem by using a version of Mahalanobis distance as the metric to be minimized. They implemented a regularized Mahalanobis metric in a two-layer self-organizing neural network that identifies departures from sphericity in a clustered data set. Later, Wang and Xia showed that Mahalanobis distance has problems with minimization in HEC (Wang and Xia 1997). Wang et al. (1997) echoed this argument and proposed a different metric for HEC.

In this chapter, we propose a new GA based algorithm for HEC. Our algorithm, called Genetic Algorithm with Regularized Mahalanobis (GARM) combines the optimization properties and rapid convergence of the Genetic K-Means algorithm with the classification accuracy of the Wang et al. method. Section 2 will review properties

of GA's and other hybrid algorithms, while Section 3 will review the development of Regularized Mahalanobis distance. Section 4 will introduce our GARM algorithm and Section 5 will demonstrate its performance on simulated data, and compare GARM's classification accuracy with the Genetic K-Means (GKM) algorithm. These differences are striking in data sets with complex covariance structure. The chapter concludes in Section 6 with a summary and discussion.

8.2 Genetic Algorithms in Cluster Analysis

First pioneered in the 1960's, GA's have been widely studied and applied in many fields of science and engineering (Holland 1975). Not only do GA's provide an alternative method to solving optimization problems, but they also consistently outperform gradient-based methods in many multi-valued optimization problems. Many of the real world problems that involve finding optimal combinations of parameters which might prove difficult for traditional methods are ideal for GA's. Some applications of GA's have included problems in engineering, economics, and game theory (Holland 1992), as well as computational sciences (Forrest 1993), and biology (Sumida et al. 1990). GA's were introduced as a computational analogy of adaptive systems. They are modeled on the principles of evolution via natural selection, employing a population of individuals that undergoes selection in the presence of variation-inducing operators such as mutation and recombination. The GA represents candidate solutions to a problem as strings of numbers. It then ranks the quality of the string solutions with a fitness function, and forces the strings to change by applying genetic operations. The strings adapt through many iterations, searching for an optimal solution to the problem. The function that ranks the quality of candidate solutions is called the fitness function. Generally speaking, the kind of fitness function that is used

in a GA depends on the problem at hand and reproductive success is proportional to fitness.

A number of researchers have implemented GA's in cluster methodology (Bhuyan et al. 1991, Jones and Beltramo 1991). Because it relies on a GA type evolution process, the GKM algorithm must define a fitness function to guide the optimization process. For K clusters, each with n_k data points, Krishna and Murty (1999) used the total within cluster variance $\mathbf{W} = \sum_{k=1}^{K} \mathbf{W}_k$ where $\mathbf{W}_k = \sum_{i=1}^{n_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \ (\mathbf{x}_i - \boldsymbol{\mu}_k)$ as the fitness function. Here, $\boldsymbol{\mu}_k$ is the centroid of cluster k. Their idea was that one measure of the quality of a clustered data set is the sum of the variances of the clusters, with smaller values denoting better clusterings. Their algorithm utilizes the global optimization properties of the genetic algorithm to find the combination of cluster assignments that minimizes the sum of cluster variances. Krishna and Murty demonstrated that their GKM algorithm calculates clusters with smaller total within cluster variance values than the traditional K-Means algorithm (Krishna and Murty 1999), leading to more compact computed clusters. The GA-string representation and associated operations efficiently searches for a global minimum in the value of total within cluster variance.

Although their method is based on GA's, Krishna and Murty do not use the standard GA operations of crossover and mutation. It is known that implementing the crossover operation in some clustering routines can be computationally expensive, especially when the fitness function depends on using the crossover operation (Bhuyan et al. 1991, Jones and Beltramo 1991). Instead of the standard random mutation operation, Krishna and Murty introduce a biased mutation operation. In their biased mutation operation, the probability of mutating to a cluster is related to the distance from the data point to the center of the cluster. The closer a data point is to a cluster center, the higher the probability of mutating to this cluster. This innovation

transformed the mutation operation from a random process into a guided search that can rapidly find the optimal combination of cluster assignments. Krishna and Murty also appealed to Markov chain theory to show that their process asymptotically converges to a global optimum in fitness value.

In addition to the reproduction and modified mutation operations, the GKM algorithm incorporates another strategy that speeds the convergence of the cluster solution, called the Genetic K-means operation. In the Genetic K-means operation, strings from the population are randomly selected and each entry in the string is assigned to the cluster with the nearest (Euclidean distance) centroid. The algorithm then recalculates the cluster centroids and reinserts the mutated strings into the population. Because the strings that undergo the Genetic K-means operation generally have among the smallest total within cluster variation, they propagate on to subsequent generations, replacing those strings whose initial assignments were far from optimal. We found that incorporating the Genetic K-means operation in the GKM clustering process can dramatically shorten the convergence time.

Another issue that arises in the GKM algorithm is that of illegal strings that are missing datapoints from one or more clusters. This leads to singular values in the definition of cluster centroids and problems with computational overhead. The GKM algorithm checks each string to see if it is illegal, and replaces empty clusters with randomly generated singleton clusters when one is found, ensuring that each of the K clusters contain at least one data point.

We can summarize of GKM clustering process as follows (Krishna and Murty 1999):

1. Initialize the population of strings that represents cluster assignments.

2. Start the reproduction loop.

3. Perform the biased mutation operation on the population of strings.

4. Perform the Genetic K-means operation on the population of strings.

5. Update the population mean values and calculate fitness scores for each string.

6. Reproduce the population based on fitness.

7. Repeat steps 3 through 6 until convergence or the maximum number of generations.

In section 4, we will show how this process and the associated operations can be adapted for HEC.

8.3 Regularized Mahalanobis Distance

Algorithms based on minimizing Euclidean distance generally compute hyperspherical clusters (Wang et al. 1997). However, many data sets have a more complex structure. Euclidean distance-based algorithms generally split elongated clusters, which leads to a higher rate of misclassification (Mao and Jain 1996). To improve their accuracy, Mao and Jain (1996) and Wang et al. (1997) used Mahalanobis distance, or its variant, in clustering algorithms. Mahalanobis distance is a generalization of Euclidean distance between two points that takes account of direction. Suppose that cluster k in a data set has centroid $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. Given an observation in the data set $\mathbf{x}_i$, we can calculate its Mahalanobis distance to the cluster's centroid as

$$D_{MD}(\mathbf{x}_i, \boldsymbol{\mu}_k) = [(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)\ ]^{1/2} \tag{8.5}$$

Mao and Jain (1996) implemented a two-layer self-organizing network to identify hyperellipsoidal clusters. The first layer had a PCA subnetwork that identifies the

hyperellipsoidal shapes, and a second layer uses a self-organizing competitive learning algorithm to refine the shape estimation. Their algorithm also relies on a Kolmogorov-Smirnov statistic to test the significance of normality in the identified clusters. Mao and Jain use Regularized Mahalanobis distance, defined as:

$$D_{RMD}(\mathbf{x}_i, \boldsymbol{\mu}_k) = (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \left[(1-\lambda)(\boldsymbol{\Sigma}_k + \epsilon\mathbf{I})^{-1} + \lambda\mathbf{I}\right](\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{8.6}$$

This definition is a compromise between Mahalanobis distance and Euclidean distance, with the degree of each controlled by the parameter λ. When $\lambda = 1$, the D_{RMD} is the squared Euclidean distance and when $\lambda = 0$, it is purely squared Mahalanobis distance. The ε term adds a small diagonal component to the covariance matrix to protect it from singularities in the matrix inversion. In their neural network, Mao and Jain use $\epsilon = 10^{-6}$, while λ decreases during the training process. The network starts the calculation with mostly Euclidean distance and gradually decreases the Euclidean component while protecting the covariance inversion.

Although it can identify complex structures, the Mao and Jain process is complicated and computationally intensive (Wang et al. 1997). Later, Wang and Xia (1997) critically reexamined Mao and Jain's metric definition. They argued that clustering algorithms based on Euclidean distance can reach a unique minimum because of the homogeneity with respect to direction, but Mahalanobis distance does not have the same unique minimum property. Wang and Xia proved that if the Sum of Squared Mahalanobis distances is used as a cost function to be minimized, then it is equal to a constant: Given a set of n patterns that belong to a cluster in d-dimensional space, if the covariance matrix $\boldsymbol{\Sigma}$ is invertible, for a cluster with mean $\boldsymbol{\mu}$, then $\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}) = d\,(n-1)$

In fact:

$$
\begin{aligned}
& \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \qquad (8.7) \\
= & \sum_{i=1}^{n} Trace[(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)] \\
= & \; Trace \left[\boldsymbol{\Sigma}_k^{-1} \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \right] \\
= & \; Trace \left[\boldsymbol{\Sigma}_k^{-1} (n-1) \boldsymbol{\Sigma}_k \right] \\
= & \; d\,(n-1)
\end{aligned}
$$

Let M_{ik} be a matrix of cluster assignments for the data set such that $\{M_{ik} \in \{0,1\} : i = 1, 2, \ldots, n; k = 1, 2, \ldots, K\}$. The squared Mahalanobis distance between data point $\mathbf{x}_i$ and cluster mean $\boldsymbol{\mu}_k$ is given as

$$
D_{MD}(\mathbf{x}_i, \boldsymbol{\mu}_k) = (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \qquad (8.8)
$$

The cost function in the clustering process is is given by

$$
E_k = \sum_{i=1}^{n} \sum_{k=1}^{K} M_{ik} D_{MD}(\mathbf{x}_i, \boldsymbol{\mu}_k) \qquad (8.9)
$$

By the previous theorem, we have that

$$
\sum_{i=1}^{n} M_{ik} D_{MD}(\mathbf{x}_i, \boldsymbol{\mu}_k) = d(n-1) \qquad (8.10)
$$

So then

$$\begin{aligned} E_k &= \sum_{i=1}^{n}\sum_{k=1}^{K} M_{ik} D_{MD}(\mathbf{x}_i, \boldsymbol{\mu}_k) \\ &= \sum_{k=1}^{K} d(n-1) \\ &= d(n-K) \end{aligned}$$

Wang and Xia go on to write that the results from Mao and Jain mainly come from the Euclidean component of the distance metric during the training process, with some hyperellipsoidal structure arising in the transition from Euclidean distance to Mahalanobis distance. In addition, Wang et al. (1997) put forth a new expression for Regularized Mahalanobis distance that overcomes the unique minimization problem.

$$D_W(\mathbf{x}_i, \boldsymbol{\mu}_k) = |\boldsymbol{\Sigma}_k|^c (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{8.11}$$

where c is a scale factor greater than zero. They show that this function is not equal to a constant, and that this expression minimizes cluster variance in all directions. Wang et al. implemented this expression as a fitness function in a GA population. They use the traditional GA operations of mutation and crossover, demonstrating that their method accurately classifies hyperellipsoidal clusters.

8.4 Genetic Algorithm with Regularized Mahalanobis Distance (GARM)

This section describes our new algorithm which can accurately and efficiently classify hyperellipsoidal data. We will describe a new regularization for the Mahalanobis distance and also propose a different fitness function based on the sum of the Regularized

Mahalanobis distances between the cluster centers and the data points. We will also show that our algorithm asymptotically converges to a global minimum.

8.4.1 Regularized Mahalanobis Distance

In their paper, Wang et al. (1997) proposed (8.11) as their Regularized Mahalanobis distance and used it as an objective function to be minimized. However, Wang et al. do not explain why this works as a metric that can identify hyperellipsoidal clusters. They also state that the exponent of the determinant c must be greater than zero, but they do not explain how this affects the outcome of the algorithm.

Another issue that Wang et al. do not address is singularities in the covariance matrix. Mao and Jain (1996) add a small regularization parameter ϵ to the diagonal of their cluster covariance matrix $\mathbf{\Sigma}_k$ to prevent singularites in the inversion process. Wang et al. do not include such a protection factor, making (8.11) susceptible to singularities in the covariance inversion.

We propose a new regularization for the Squared Mahalanobis distance between data point $\mathbf{x}_i$ and cluster mean $\boldsymbol{\mu}_k$ to be

$$D_{WB}(\mathbf{x}_i, \boldsymbol{\mu}_k) = |\mathbf{\Sigma}_k|^{\frac{1}{2}} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\mathbf{\Sigma}_k + \epsilon \mathbf{I})^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{8.12}$$

where $\boldsymbol{\mu}_k$ and $\mathbf{\Sigma}_k$ are the mean and covariance matrix of cluster k respectively, and ϵ is a small value that protects the inversion process from singularities. We choose the exponent of the covariance determinant to be $\frac{1}{2}$ because $|\mathbf{\Sigma}|^{\frac{1}{2}}$ is the square root of the generalized variance of multivariate data set. Our numerical trials have shown that this exponent accurately classified complex hyperellisoidal data, while other values gave less favorable results. In addition, in equation (8.12), since $(\mathbf{\Sigma}_k + \epsilon \mathbf{I})$ is invertible, our expression for the Regularized Mahalanobis distance is a special case of (8.11).

Therefore, it is not equal to a constant and can be used as a minimized objective function.

Using the Woodbury formula for matrices $\mathbf{A}, \mathbf{U}$, and $\mathbf{V}$:

$$\left(\mathbf{A}+\mathbf{U}\mathbf{V}^T\right)^{-1} = \mathbf{A}^{-1} - \left[\mathbf{A}^{-1}\mathbf{U}\left(\mathbf{I}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^T\mathbf{A}^{-1}\right] \tag{8.13}$$

If we let $\mathbf{A}=\mathbf{\Sigma}$, $\mathbf{U}=\epsilon\mathbf{I}$, and $\mathbf{V}=\mathbf{I}$, then we have the following:

$$\begin{aligned}
(\mathbf{\Sigma}+\epsilon\mathbf{I})^{-1} &= \left(\mathbf{\Sigma}+(\epsilon\mathbf{I})\mathbf{I}^T\right)^{-1} \\
&= \mathbf{\Sigma}^{-1} - \left[\mathbf{\Sigma}^{-1}(\epsilon\mathbf{I})\left(\mathbf{I}+\mathbf{I}^T\mathbf{\Sigma}^{-1}(\epsilon\mathbf{I})\right)^{-1}\mathbf{I}^T\mathbf{\Sigma}^{-1}\right] \\
&= \mathbf{\Sigma}^{-1} - \left[\mathbf{\Sigma}^{-1}\epsilon\left(\mathbf{I}+\epsilon\mathbf{\Sigma}^{-1}\right)^{-1}\mathbf{\Sigma}^{-1}\right]
\end{aligned} \tag{8.14}$$

We also note that

$$\begin{aligned}
\left(\mathbf{I}+\epsilon\mathbf{\Sigma}^{-1}\right)^{-1} &= \left(\mathbf{I}\left(1+\epsilon\right)\mathbf{\Sigma}^{-1}\right)^{-1} \\
&= \left(\left(1+\epsilon\right)\mathbf{I}\mathbf{\Sigma}^{-1}\right)^{-1} \\
&= \frac{1}{1+\epsilon}\mathbf{\Sigma}
\end{aligned} \tag{8.15}$$

So, that gives

$$\begin{aligned}
(\mathbf{\Sigma}+\epsilon\mathbf{I})^{-1} &= \mathbf{\Sigma}^{-1} - \left[\mathbf{\Sigma}^{-1}\frac{\epsilon}{1+\epsilon}\mathbf{\Sigma}\mathbf{\Sigma}^{-1}\right] \\
&= \mathbf{\Sigma}^{-1} - \left[\frac{\epsilon}{1+\epsilon}\mathbf{\Sigma}^{-1}\right] \\
&= \left(1-\frac{\epsilon}{1+\epsilon}\right)\mathbf{\Sigma}^{-1}
\end{aligned} \tag{8.16}$$

So for $\epsilon \to 0$, then $\left(1-\frac{\epsilon}{1+\epsilon}\right)\mathbf{\Sigma}^{-1} \approx \mathbf{\Sigma}^{-1}$

In order to use our Regularized Mahalonobis expression in a clustering algorithm, we need to define a cost function to be minimized. Let M_{ik} be a matrix of cluster assignments for the data set such that $\{M_{ik} \in \{0,1\} : i = 1,2,\ldots,n; k = 1,2,\ldots,K\}$. Let $n_k = \sum_{i=1}^{n} M_{ik}$ be the sum of patterns in the ith cluster. We can express the Regularized Mahalanobis distance between data point $\mathbf{x}_i$ and associated cluster mean by

$$D_{GARM}(\mathbf{x}_i, \boldsymbol{\mu}_k) = |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T(\boldsymbol{\Sigma}_k + \epsilon\mathbf{I})^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k) \tag{8.17}$$

The cost function in the clustering process is defined by

$$E_k = \sum_{i=1}^{n}\sum_{k=1}^{K} M_{ik} D_{GARM}(\mathbf{x}_i, \boldsymbol{\mu}_k) \tag{8.18}$$

This function becomes the Sum of Within Cluster Generalized Variances of the cluster set. Our clustering algorithm will try to iteratively minimize this expression with respect to cluster assignments.

8.4.2 Clustering with the Genetic Algorithm

In order to use (8.18) in a clustering algorithm, we must search for a minimum in the metric. Clustering metrics generally have high dimensionality and are non-linear, so one should implement a method that quickly searches for a global minimum. Following the example of Krishna and Murty (1999), we structured our clustering algorithm as a GA. For the n_k points that belong to cluster k, let $\mathbf{W}_k = \sum_{i=1}^{n_k} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T(\boldsymbol{\Sigma}_k + \epsilon\mathbf{I})^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)$. We then defined the fitness function of the GA to be $\mathbf{W} = \sum_{k=1}^{K} \mathbf{W}_k$. This fitness function is the sum of generalized variances of the clusters. We use this sum of regularized Mahalanobis distances in the same way that Krishna and Murty use the Total Within Cluster Variance in GKM.

Our GARM algorithm relies on analogous GA-hybrid operations as the GKM method, giving our algorithm the same level of computational complexity as Krishna and Murty's. This implementation is an improvement over the method proposed by Wang et al. (1997). Wang et al. relied on the traditional GA operations of crossover, mutation, and reproduction. Not only do the crossover operations increase the computational complexity of their algorithm, but clustering with the Wang et al. implementation can take hundreds to thousands of iterations to converge. Simulations will later show that our method quickly converges to a global optimum.

We wrote the algorithm in the MATLAB language and ran trials on a Pentium III computer with Windows XP. Parameters that the user can control are the size of the population, the mutation probability, and the termination criterion. We used the *string-of-group numbers* coding (Jones and Beltramo 1991) where each data point in the chromosome string can assume values $\{1, 2, \ldots, K\}$, denoting the cluster to which the data point belongs. While the population undergoes genetic operations, illegal strings can be formed that are missing data points from one or more clusters. We handled illegal strings by randomly inserting singleton clusters to ensure that each string has at least one data point assigned to each cluster.

Let the algorithm have n data points to be partitioned into K clusters, and mutation probability *mutprob*. The following describes the steps in our GARM algorithm along with the pseudocode for each step.

1. Initialization: GARM first randomly generates cluster assignments in each string. The initial random cluster assignments are drawn from a uniform probability distribution, so each string has approximately $\frac{n}{K}$ members assigned to each clusters.

   Pseudocode:

   Start

for $i = 1$ to N

for $j = 1$ to n

$population(i, j) = integer(rand * K)$

end

end

end

2. Mutation: In a GA framework, mutation introduces random variability into a population that mimics random changes in genetic structure found in organisms. In a GA based clustering situation, the mutation operation changes the value of a cluster assignment according to some probability distribution. Our mutation operation is analogous to the GKM biased mutation operation, except that we use the Regularized Mahalanobis metric to gauge the distance from the cluster centroid to the data point. The algorithm randomly selects population strings and data points to undergo mutation. After making the selection, the probability of a point mutating into a given cluster assignment is related to the distance between the data point and the cluster centroid. The closer a data point is to a cluster centroid, the higher the probability that it will mutate into that cluster. Let $d_j = d(\mathbf{x}_i, \boldsymbol{\mu}_k)$ denote our Regularized Mahalanobis distance from data point $\mathbf{x}_i$ to cluster center $\mathbf{c}_j$. The probability of mutating into the respective cluster assignments is given by

$$p_j = \frac{c_m d_{\max} - d_j}{\sum_{i=1}^{K} (c_m d_{\max} - d_i)}$$

where c_m is a constant and $d_{\max} = \max\{d_j\}$. In our case, we set $c_m = 1$.

Pseudocode:

Start

Randomly Select population member

Randomly generate points $\{x_1, x_2, \ldots, x_r\}$ that undergo mutation by using *mutprob*

For $i = 1$ to r

For $j = 1$ to K

compute d_j

end

$p_j = \frac{c_m d_{\max} - d_j}{\sum_{i=1}^{K} (c_m d_{\max} - d_i)}$

Generate new cluster assignments for x_i according to probabilities p_j.

end

Check to see if string is illegal.

Insert mutated string into population.

end

3. Mahalanobis Operation: This operation randomly selects a string from the population and then assigns each point in the string to the cluster that has the closest Regularized Mahalanobis distance. This operation is analogous to the K-means Operation in the GKM algorithm. Because strings that undergo this Mahalanobis operation generally have among the lowest Sum of Generalized Variance in the population, they propagate on to subsequent generations. Including this operation greatly speeds convergence of the algorithm to a solution. Let $d_j = d(\mathbf{x}_i, \mathbf{c}_j)$ denote our Regularized Mahalanobis distance from data

point $\mathbf{x}_i$ to cluster center $\mathbf{c}_j$. We can summarize this Mahalanobis operation as follows.

Pseudocode:

Start

Randomly Select population member

For $i = 1$ to n

for $j = 1$ to K

$d_j = d(\mathbf{x}_i, \mathbf{c}_j)$

end

$x_i = find(d_j = \min(d_j))$

end

Check to see if string is illegal.

Insert mutated string into population.

end

4. Selection: GARM uses the same kind of reproduction strategy as GKM. The Sum of Generalized Variance becomes the fitness of the string $F(s_i)$. The probability of reproduction then is given by

$$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}$$

Members of the next generation are selected according to their relative fitness values in a roulette wheel selection method. We include the same kind of σ-truncation method as Krishna and Murty (1999).

Pseudocode:

Start

for $i = 1$ to N

$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}$

Check σ-truncation of fitness values

end

Select member of next generation according to probabilities $P(s_i)$

end

8.4.3 Convergence

Rudolph (1997) proved that the canonical GA converges to a global optimum in the fitness value. Krishna and Murty (1999) also proved that the GKM algorithm converges to a global minimum in the total within cluster variance. Our GARM algorithm, which uses the same GA structure, also converges to a global optimum in fitness. GARM uses the mutation and Genetic Mahalanobis operation in the same way that Krishna and Murty use their operations. The only difference is the fitness definition and the metric used to measure the distance between the observations and the means of the clusters. The algorithm converges in a similar way as Krishna and Murty's algorithm since the fitness function and the metric used in GARM does not affect the behavior of the Markov chain established in the state space. The state space describes the populations containing legal strings, that is strings representing partitions with K nonempty clusters. From the definition of GARM, $P(t+1)$ can be determined completely by $P(t)$, where $\{P(t)\}_{t=0}$ represents the population maintained by GARM at generation t, so $\{P(t)\}_{t=0}$ is a Markov chain. The transition probabilities $p_{ij}(t) = P(P(t) = p_j | P(t-1) = p_i)$ are independent of time. This shows

that $\{P(t)\}_{t=0}$ is a time-homogeneous finite Markov Chain and that the transition matrix (p_{ij}) is the stochastic matrix of the Markov chain.

8.5 Analysis

In this section, we compared the performance of our GARM algorithm with GKM on simulated data of increasing complexity. For the simulated data, we choose different schemes, where clusters are ellipsoidal and may overlap. The automated clustering process becomes especially challenging when clusters overlap. In the plots for these examples, different colors and symbols denote cluster memberships. Black plus signs (+) mark the positions of the respective cluster means.

8.5.1 Example 1

The first example of the algorithm used 5 simulated normally-distributed bivariate clusters. This data set had 500 data points, with 100 data points assigned to each cluster. The true classification of the simulated data is given in figure 8.1.

When we applied our GARM algorithm to this data set, the successful classification rate was 95.2%. Figure 8.2 gives the classification results of our GARM algorithm on this data set.

As a comparison, we also classified the same simulated data set with GKM. Figure 8.3 shows the classification results of this trial. The classification accuracy rate for GKM was 90.2%.

Tables 8.1 and 8.2 compare the means and covariance estimates of the clusters processed with GARM and GKM respectively.

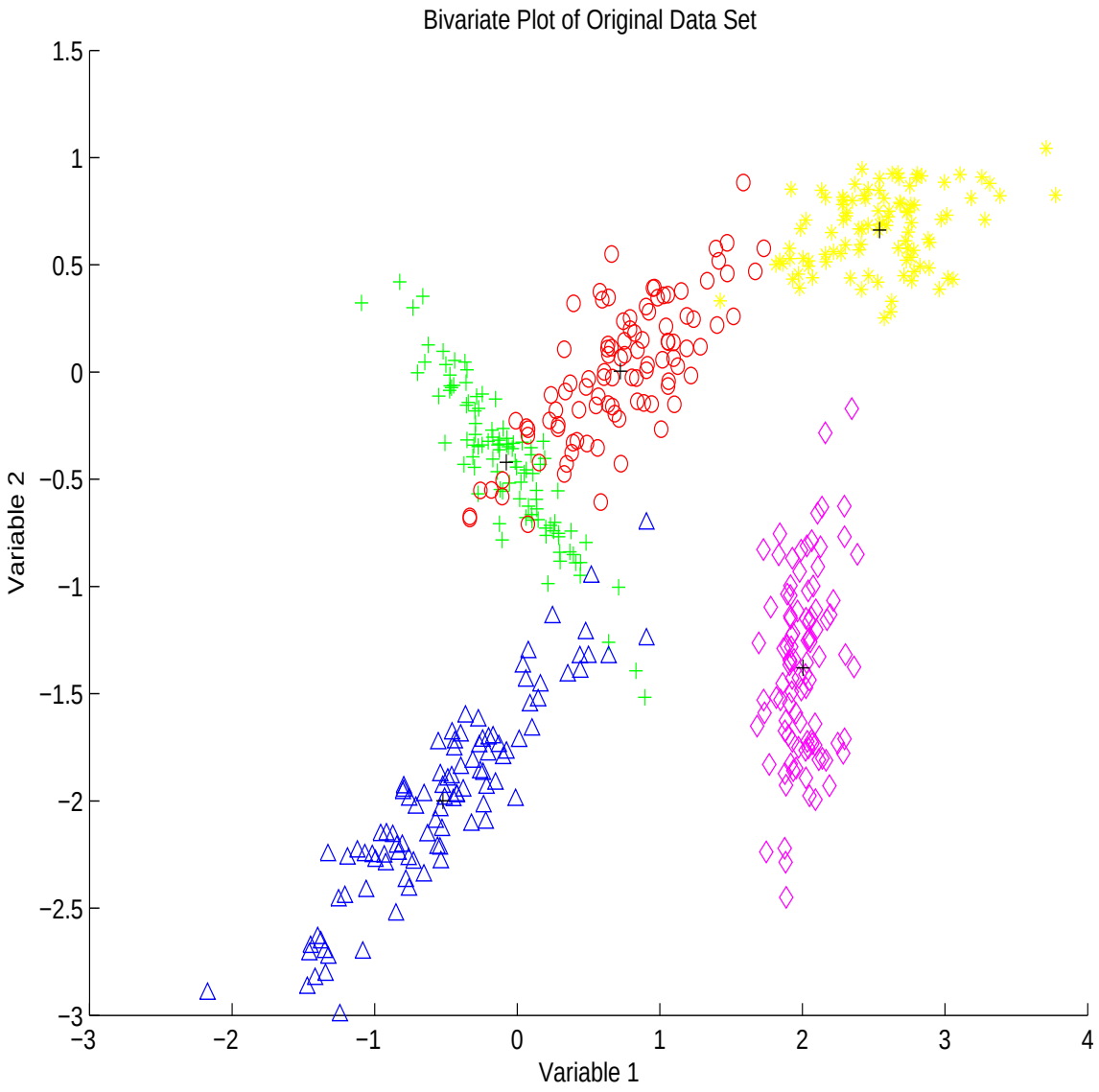

Figure 8.1: True Classification of Simulated Data Set 8-1.

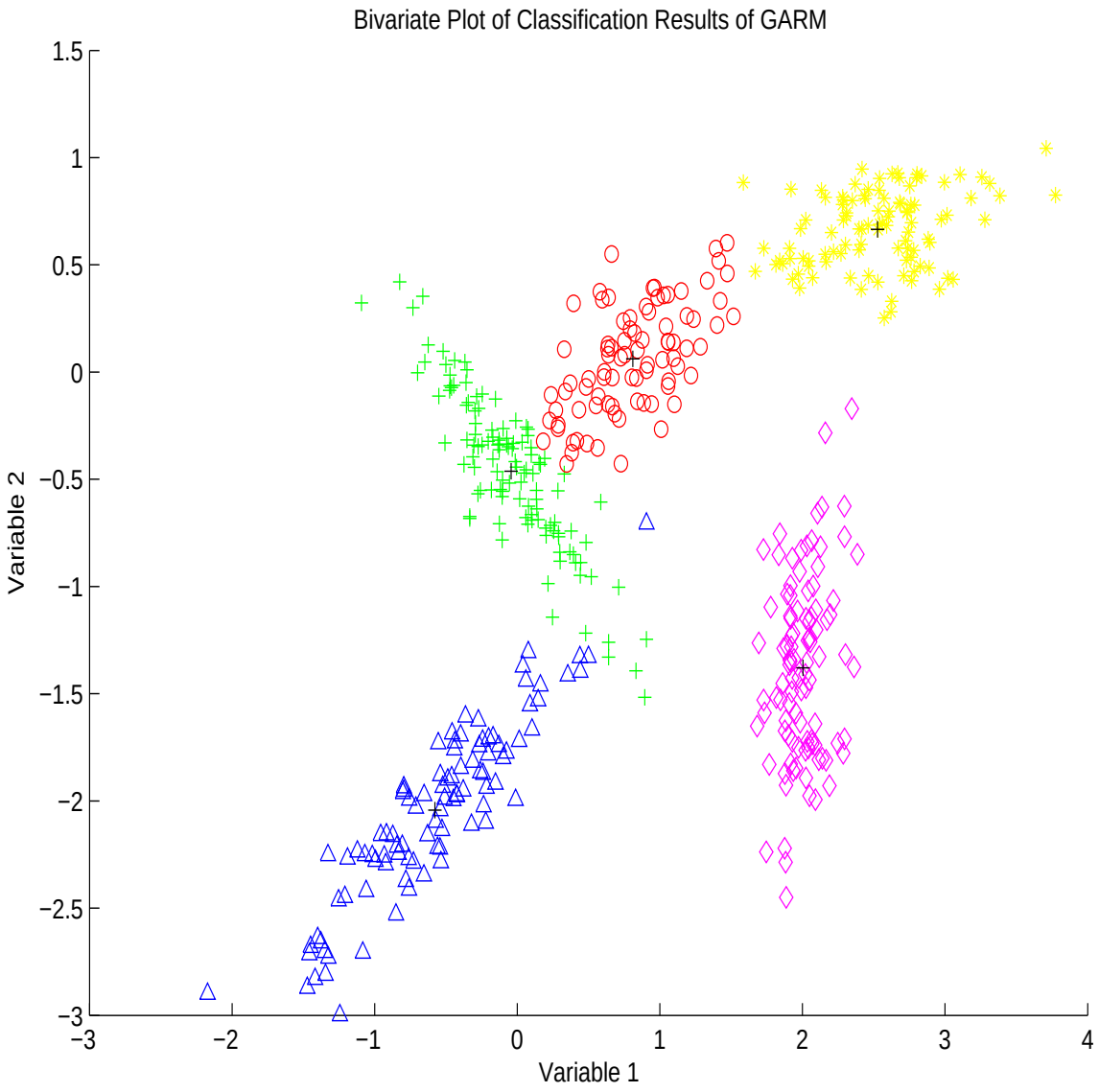

Figure 8.2: Classification Results of GARM on Simulated Data Set 8-1.

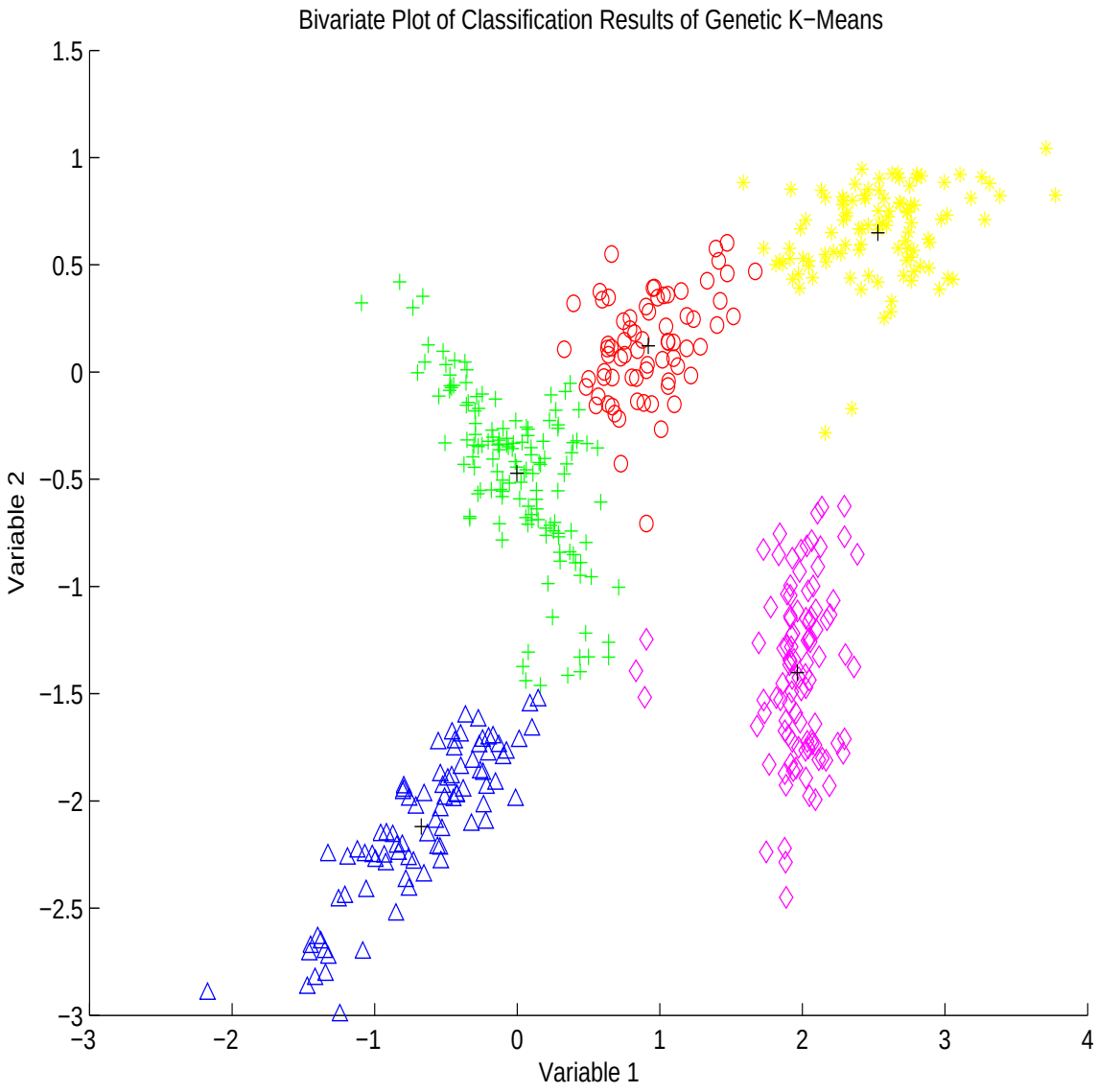

Figure 8.3: Classification Results of GKM on Simulated Data Set 8-1.

Table 8.1: Comparisons of the Mean Estimations of GARM and GKM for Different Clusters in Simulated Data Set 8-1.

Cluster	Original	GARM	GKM
1	$(-0.1, -0.4)$	$(-0.1, -0.4)$	$(0.0, -0.5)$
2	$(0.7, 0.0)$	$(0.7, 0.0)$	$(0.9, 0.1)$
3	$(-0.5, -2.0)$	$(-0.5, -2.0)$	$(-0.7, -2.1)$
4	$(2.5, 0.7)$	$(2.5, 0.7)$	$(2.5, 0.7)$
5	$(2.0, -1.4)$	$(2.0, -1.4)$	$(2.0, -1.4)$

Table 8.2: Comparisons of the Covariance Estimations of GARM and GKM for Different Clusters in Simulated Data Set 8-1.

Cluster	Original	GARM	GKM
1	$\begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.1 \end{pmatrix}$
2	$\begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0.0 \\ 0.1 & 0.1 \end{pmatrix}$
3	$\begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.3 & 0.2 \\ 0.2 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$
4	$\begin{pmatrix} 0.2 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}$
5	$\begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.1 & -0.1 \\ -0.1 & 0.1 \end{pmatrix}$

We can also compare the convergence rates of the Wang et al. method and GARM on this data set. Figures 8.4 and 8.5 show the respective convergence rates of these algorithms.

The clusters in this data are not strongly overlapped and not all of the clusters show strong ellipsoidal shapes. Because of the relatively simple structure of this data set, this case is not especially difficult to cluster. These results show that, even on this simple data set, GARM had a higher correct classification rate than GKM. In addition, GARM gave estimations of the cluster means and covariances closer to the true values than GKM. In addition, GARM converged after only 7 iterations to obtain

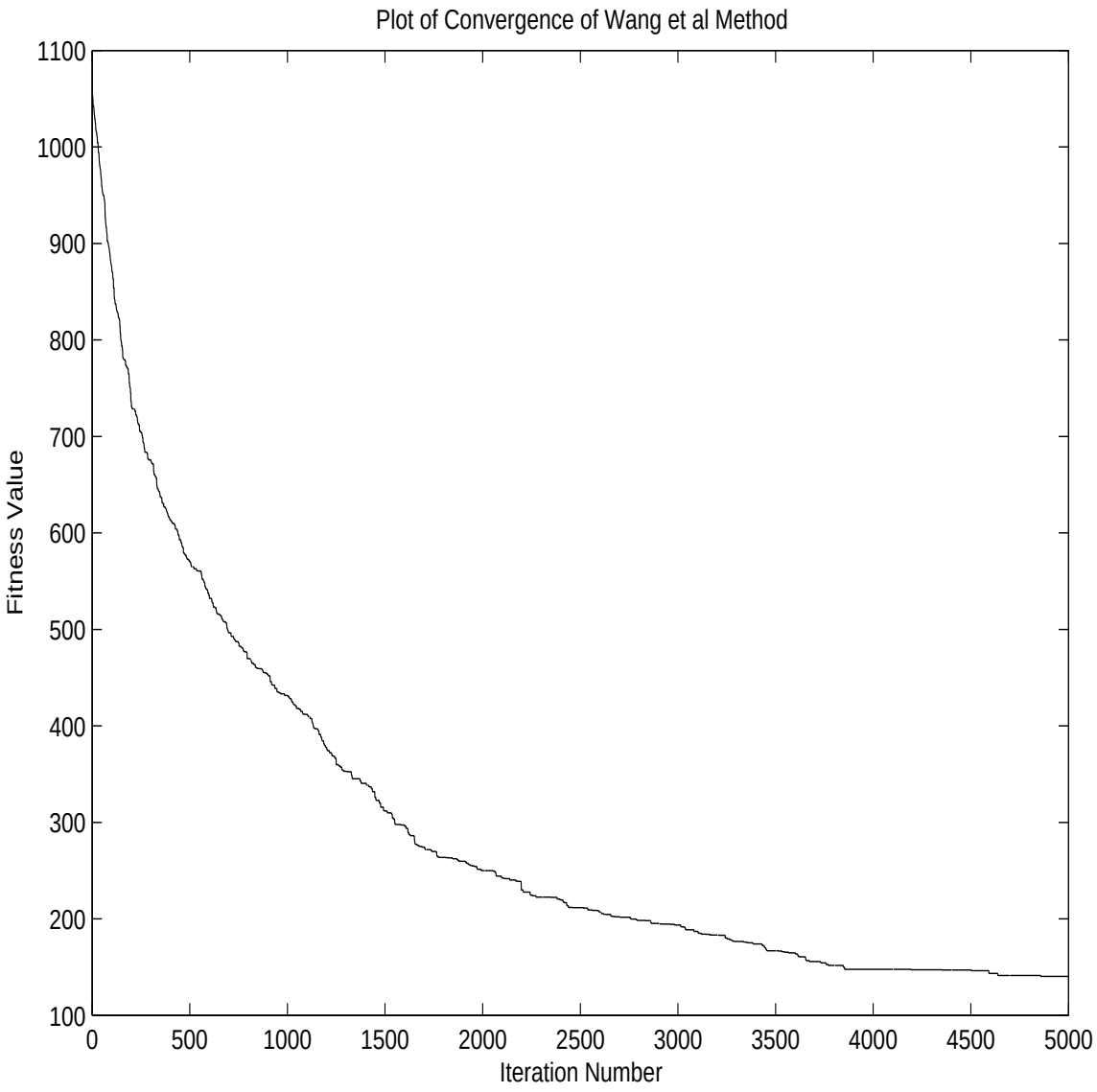

Figure 8.4: Convergence of Wang et al. Method for Simulated Data Set 8-1.

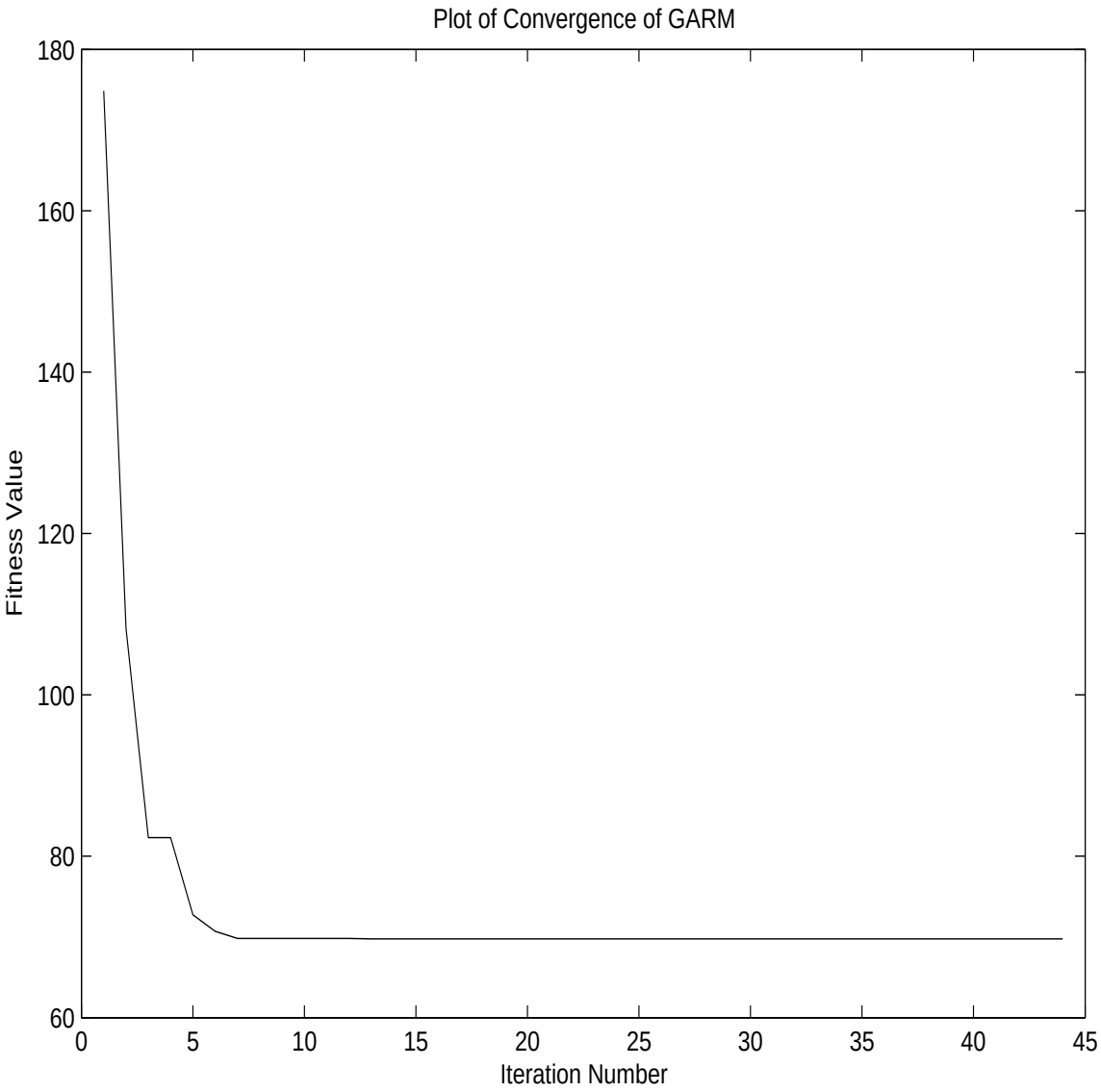

Figure 8.5: Convergence of GARM for Simulated Data Set 8-1.

an optimal classification, while the Wang et al. method took thousands of iterations to converge to a similar cluster calculation with this data set.

8.5.2 Example 2

The second simulated data are 500 bivariate points expressed in three clusters. The first cluster has 150 data points while the second cluster has 250 data points and the third cluster has 100 data points. These clusters do not overlap, but they show strong ellipsoidal structure, making Euclidian distance based classification difficult. Figure 8.6 shows the original classification of this data set.

When we applied our GARM algorithm to this data set, the successful classification rate was 100%. Figure 8.7 gives the classification results of GARM algorithm on this data set.

As a comparison, we also applied GKM to this data set. Figure 8.8 shows the classification results of this trial. The accuracy rate of GKM was 80.2%.

Tables 8.3 and 8.4 show the cluster means and covariance estimations for the respective algorithms.

We can also compare the convergence rates of the Wang et al. method and GARM on this data set. Figures 8.9 and 8.10 show the respective convergence of these algorithms.

Table 8.3: Comparisons of the Mean Estimations of GARM and GKM for Different Clusters in Simulated Data Set 8-2.

Cluster	Original	GARM	GKM
1	(2.1, 2.1)	(2.1, 2.1)	(2.1, 2.1)
2	(-4.0, -2.0)	(-4.0, -2.0)	(-3.1, -3.7)
3	(-3.0, 2.9)	(-3.0, 2.9)	(-4.2, 2.0)

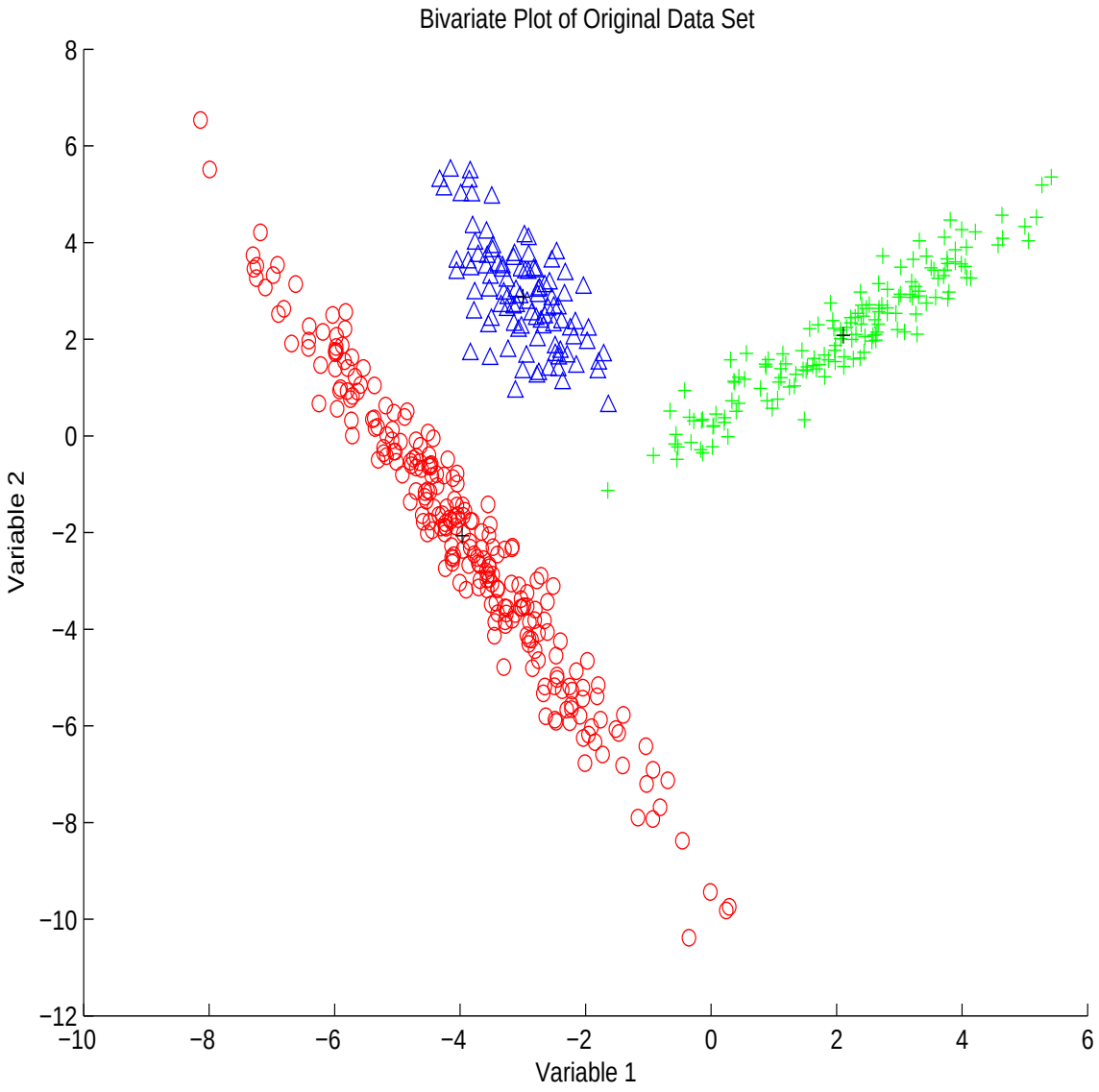

Figure 8.6: True Classification of Simulated Data Set 8-2.

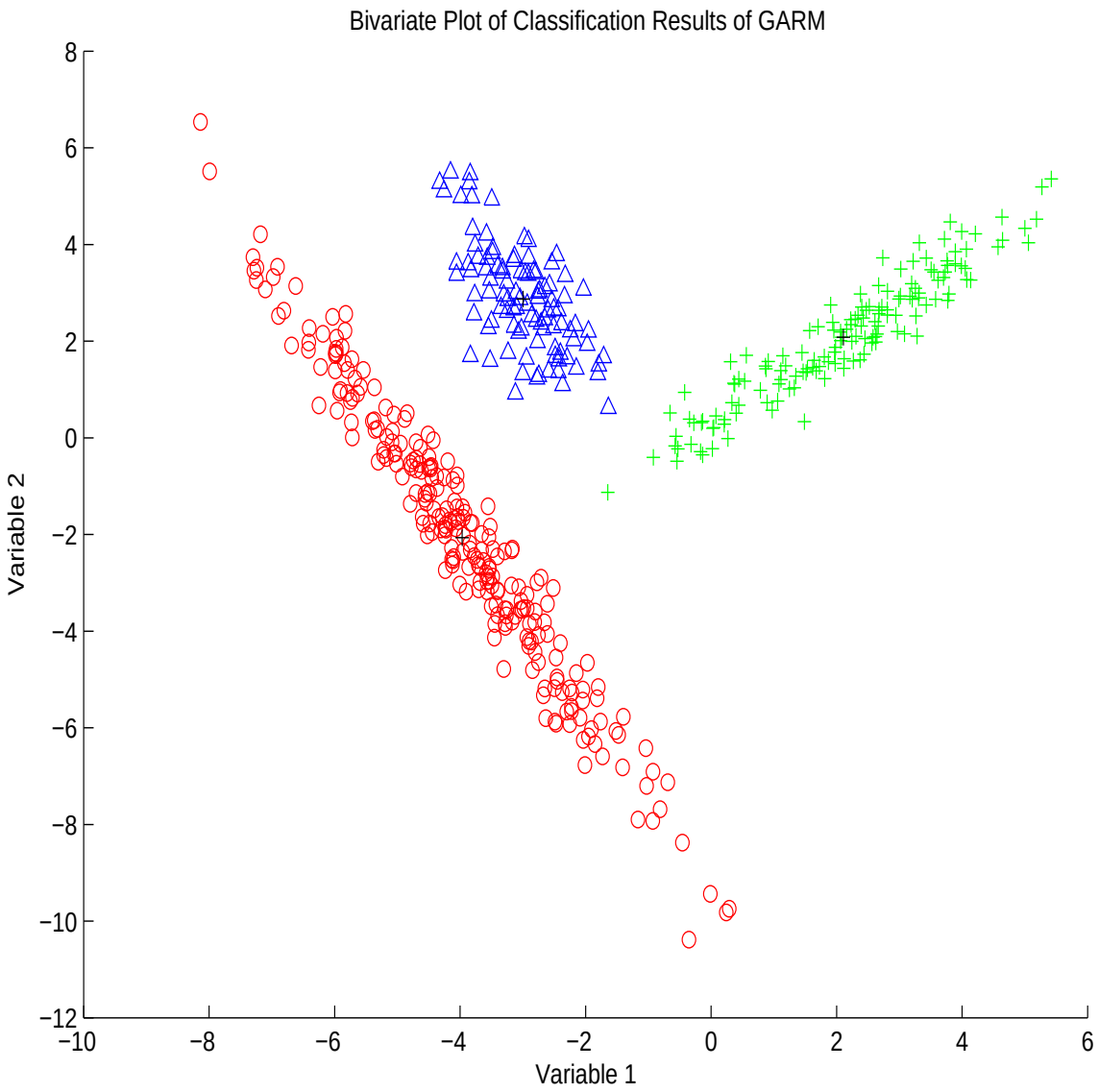

Figure 8.7: Classification Results of GARM on Simulated Data Set 8-2.

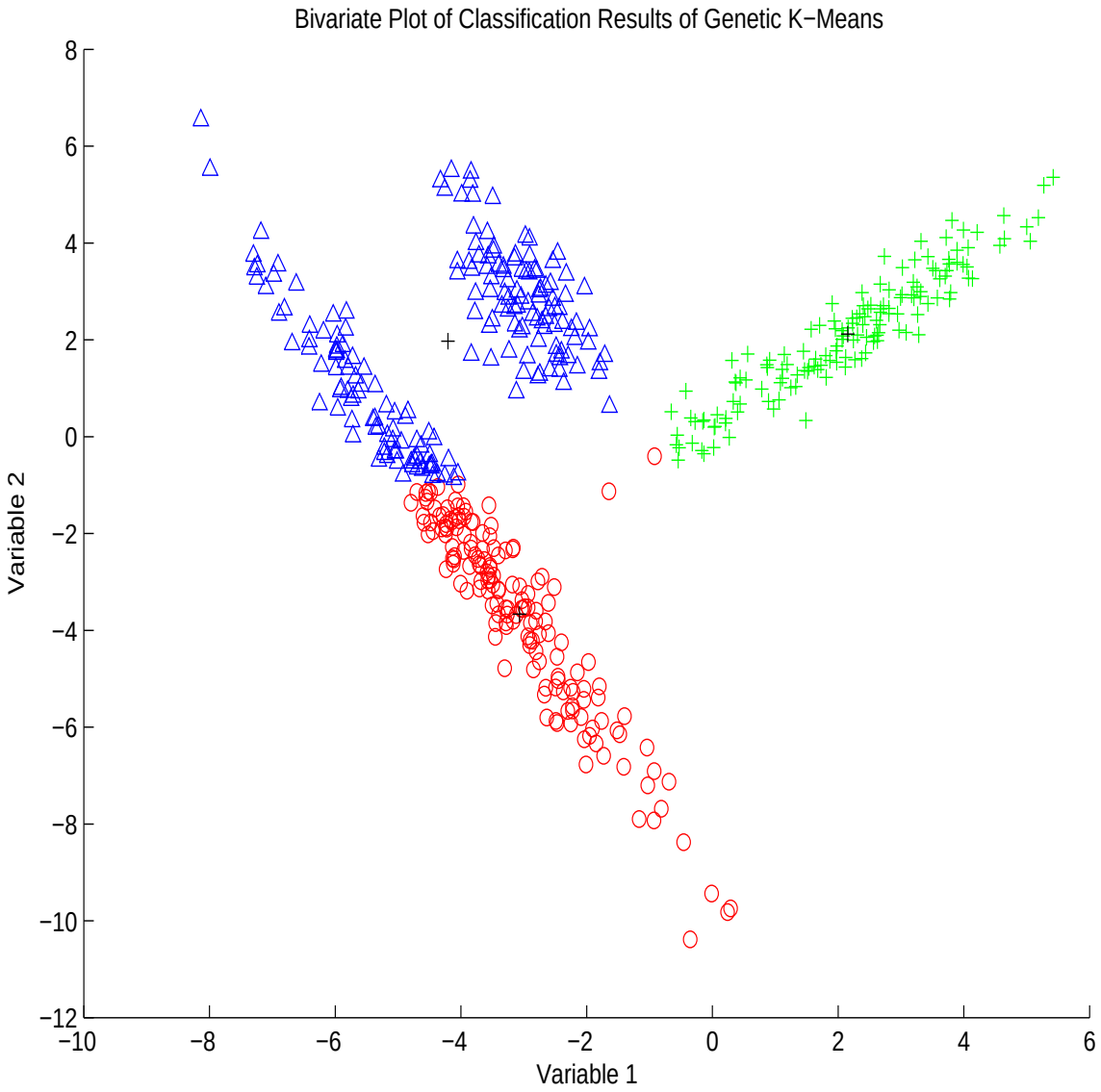

Figure 8.8: Classification Results of GKM on Simulated Data Set 8-2.

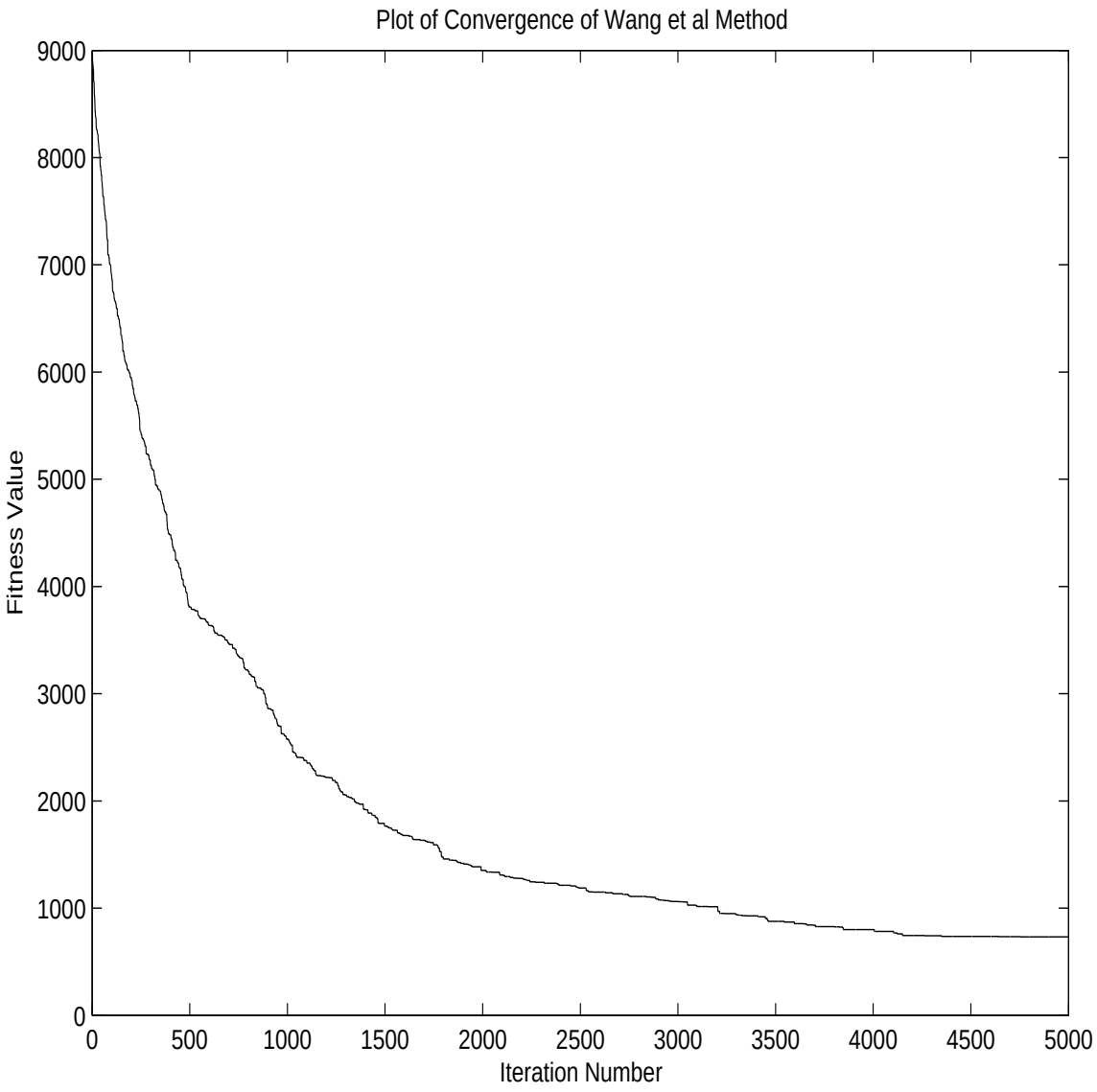

Figure 8.9: Convergence of Wang et al. Method for Simulated Data Set 8-2.

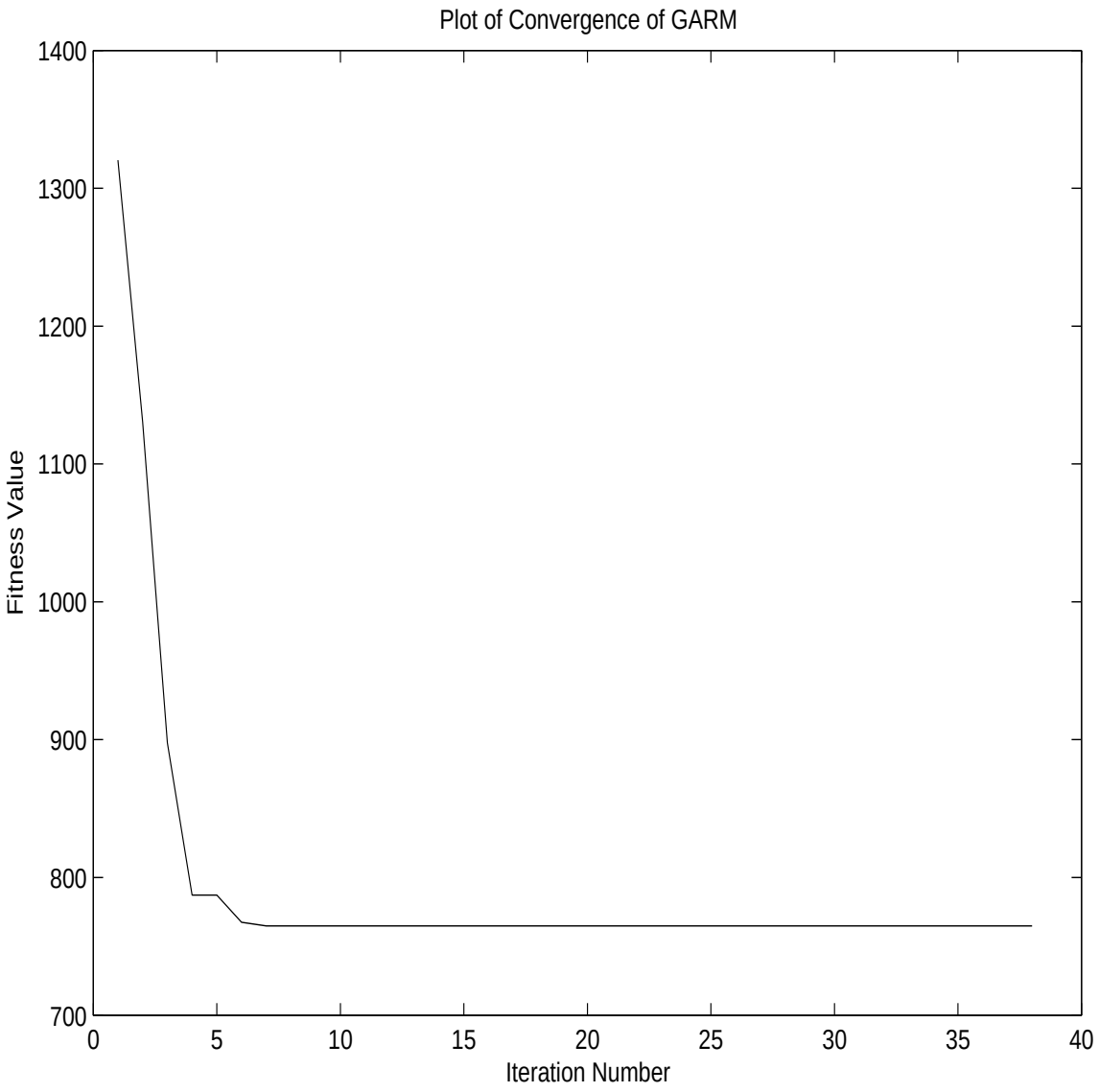

Figure 8.10: Convergence of GARM for Simulated Data Set 8-2.

Table 8.4: Comparisons of the Covariance Estimations of GARM and GKM for Different Clusters in Simulated Data Set 8-2.

Cluster	Original	GARM	GKM
1	$\begin{pmatrix} 2.2 & 1.9 \\ 1.9 & 1.7 \end{pmatrix}$	$\begin{pmatrix} 2.2 & 1.9 \\ 1.9 & 1.7 \end{pmatrix}$	$\begin{pmatrix} 2.1 & 1.7 \\ 1.7 & 1.6 \end{pmatrix}$
2	$\begin{pmatrix} 2.5 & -4.4 \\ -4.4 & 8.4 \end{pmatrix}$	$\begin{pmatrix} 2.5 & -4.4 \\ -4.4 & 8.4 \end{pmatrix}$	$\begin{pmatrix} 1.2 & -2.0 \\ -2.0 & 4.0 \end{pmatrix}$
3	$\begin{pmatrix} 0.4 & -0.4 \\ -0.4 & 1.7 \end{pmatrix}$	$\begin{pmatrix} 0.4 & -0.4 \\ -0.4 & 1.7 \end{pmatrix}$	$\begin{pmatrix} 2.3 & 0.4 \\ 0.4 & 2.7 \end{pmatrix}$

These results demonstrate that GARM is more well-suited to classifying this hyperellipsoidal data. GARM retrieved exact values of cluster means and covariances. This again showed that GARM converged quickly, after only 7 iterations, while the Wang et al. method took much longer converge to a similar cluster calculation.

8.5.3 Example 3

For the next trial, we used a simulated data set that was not so well separated, making the classification process a bigger challenge. This was again a 500 point bivariate data set expressed in three clusters. The clusters had 150, 250, and 100 data points respectively.

These clusters overlap and show strong ellipsoidal structure. The true classification of the simulated data is given in figure 8.11. When we applied our GARM algorithm to this data set, the successful classification rate was 93.2%. Figure 8.12 gives the classification results of GARM algorithm on this data set. Figure 8.13 shows the classification results of GKM trial. The GKM classification accuracy rate was 48.8%.

Tables 8.5 and 8.6 show the cluster means and covariance estimations from GARM and GKM.

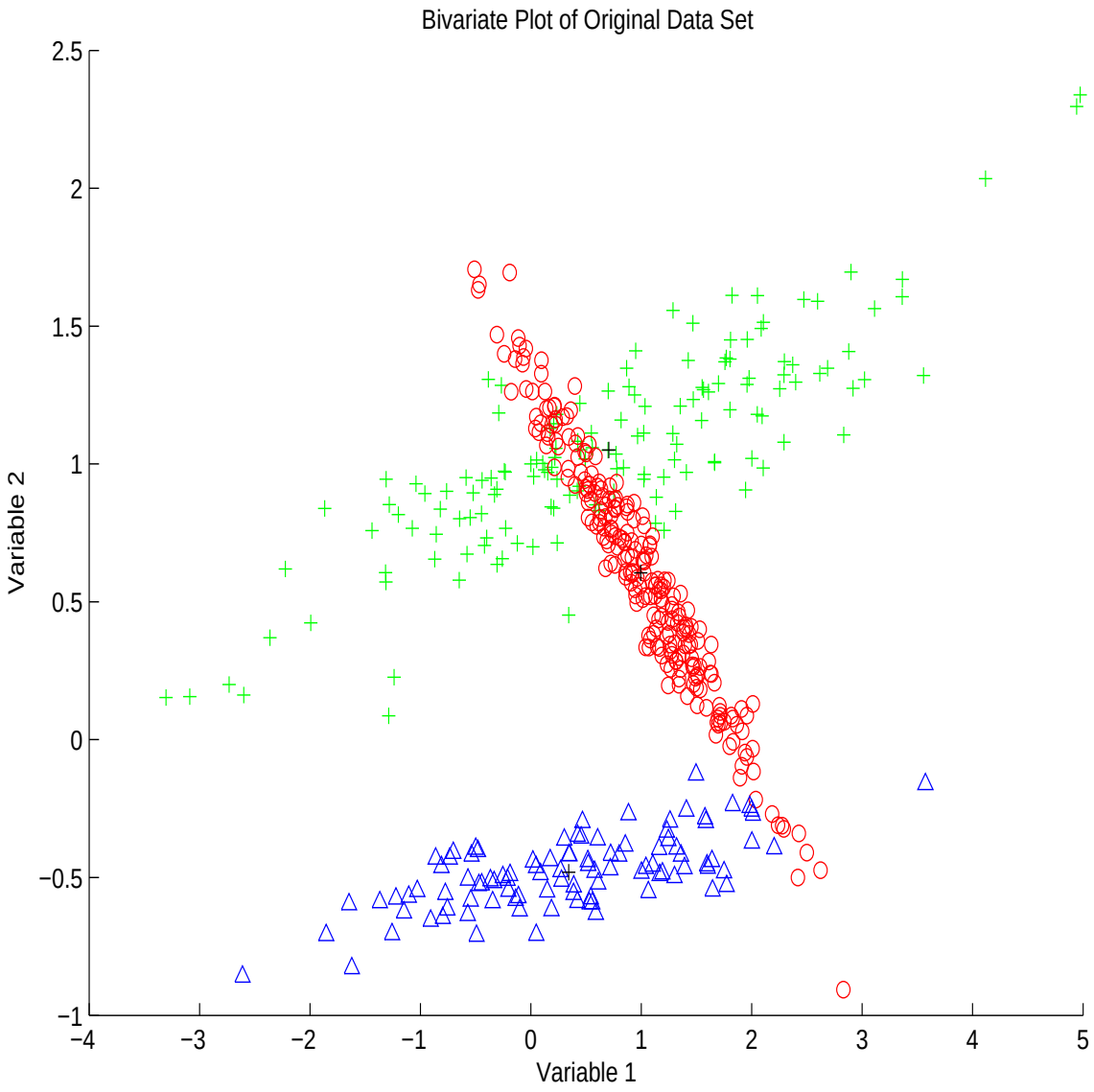

Figure 8.11: True Classification of Simulated Data Set 8-3.

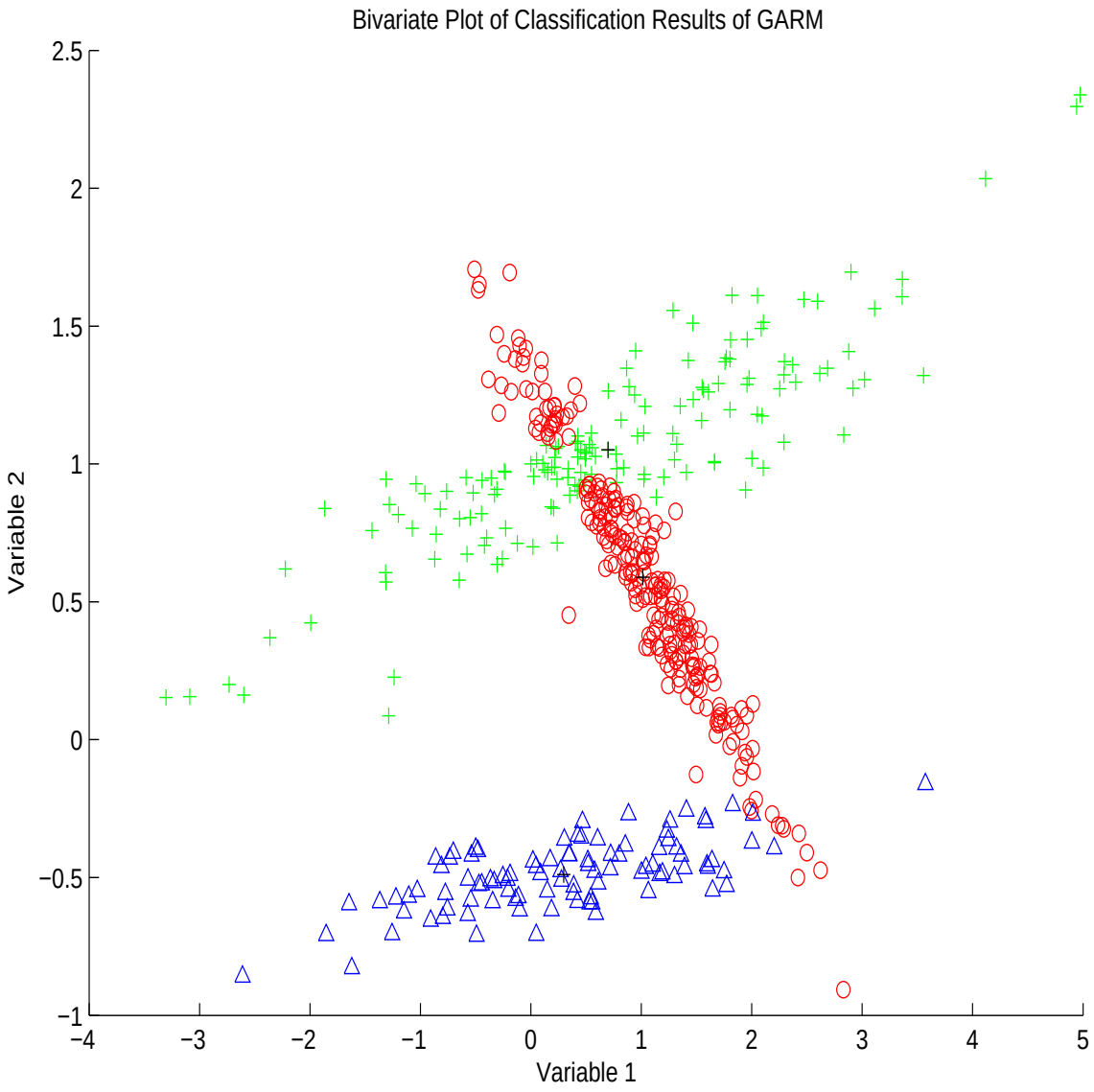

Figure 8.12: Classification Results of GARM on Simulated Data Set 8-3.

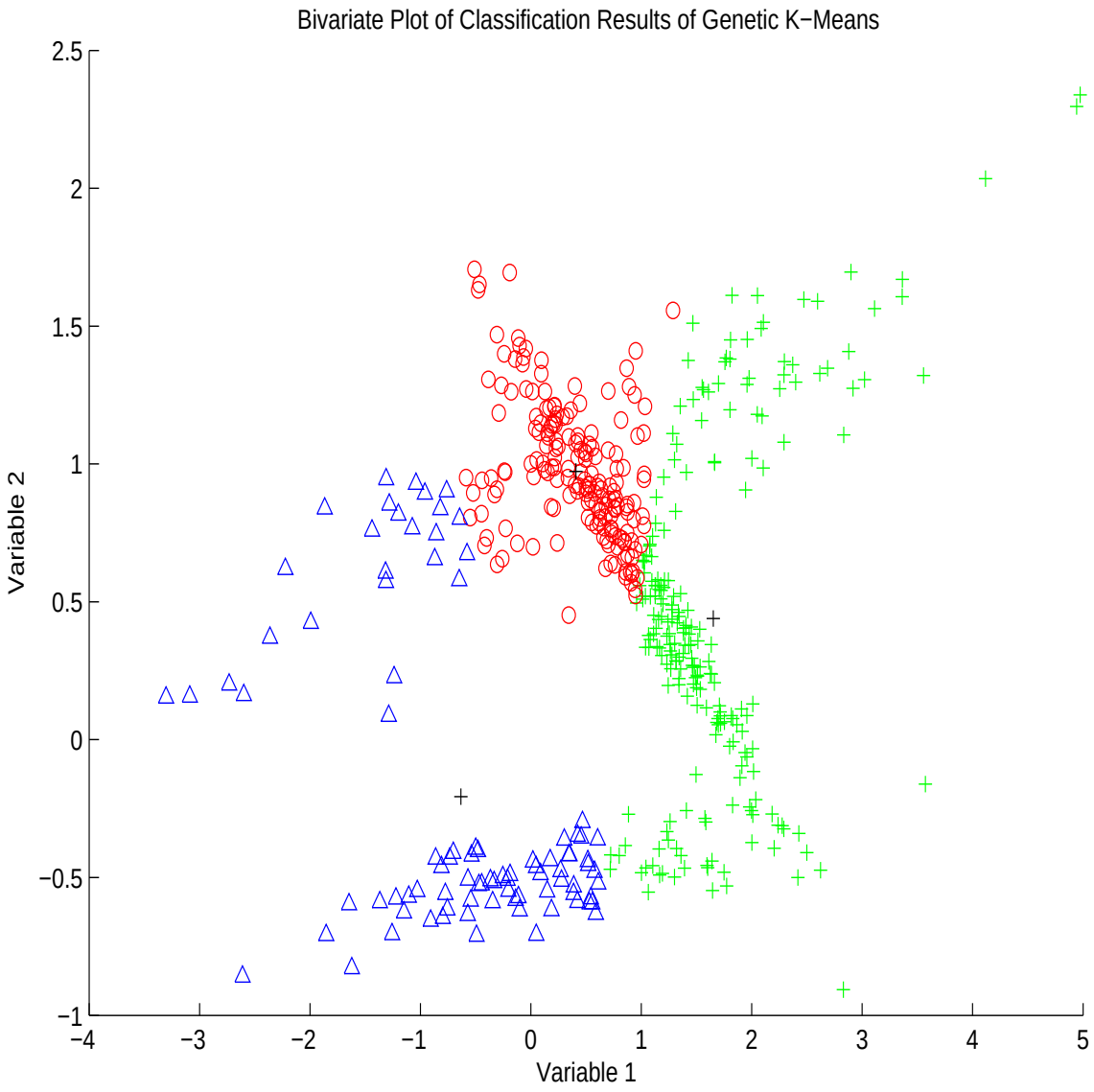

Figure 8.13: Classification Results of GKM on Simulated Data Set 8-3.

Table 8.5: Comparisons of the Mean Estimations of GARM and GKM for Different Clusters in Simulated Data Set 8-3.

Cluster	Original	GARM	GKM
1	(0.7, 1.0)	(0.7, 1.0)	(1.7, 0.4)
2	(1.0, 0.6)	(1.0, 0.6)	(0.4, 1.0)
3	(0.3, -0.5)	(0.3, -0.5)	(-0.6, -0.2)

Table 8.6: Comparisons of the Covariance Estimations of GARM and GKM for Different Clusters in Simulated Data Set 8-3.

Cluster	Original	GARM	GKM
1	$\begin{pmatrix} 2.3 & 0.5 \\ 0.5 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 2.2 & 0.5 \\ 0.5 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.4 & 0.2 \\ 0.2 & 0.4 \end{pmatrix}$
2	$\begin{pmatrix} 0.4 & -0.3 \\ -0.3 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.4 & -0.3 \\ -0.3 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0.0 \\ 0.0 & 0.1 \end{pmatrix}$
3	$\begin{pmatrix} 1.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 1.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 0.8 & -0.2 \\ -0.2 & 0.3 \end{pmatrix}$

We can again compare the convergence rates of the Wang et al. method and GARM on this data set. Figures 8.14 and 8.15 show the respective convergences of these algorithms.

This example shows a drastic difference in the classification accuracy of GARM and GKM, confirming that our algorithm is well suited to clustering these complex hyperellipsoidal data. The classification accuracy of GARM led to good cluster mean and covariance estimations, whereas GKM gives poor estimates of the means and covariance matrices. This example again demonstrated that GARM converged quickly, after only 13 iterations.

8.6 Conclusion

In this chapter, we proposed a GA clustering method based on a new Regularized Mahalanobis metric and demonstrated that the algorithm can separate complex

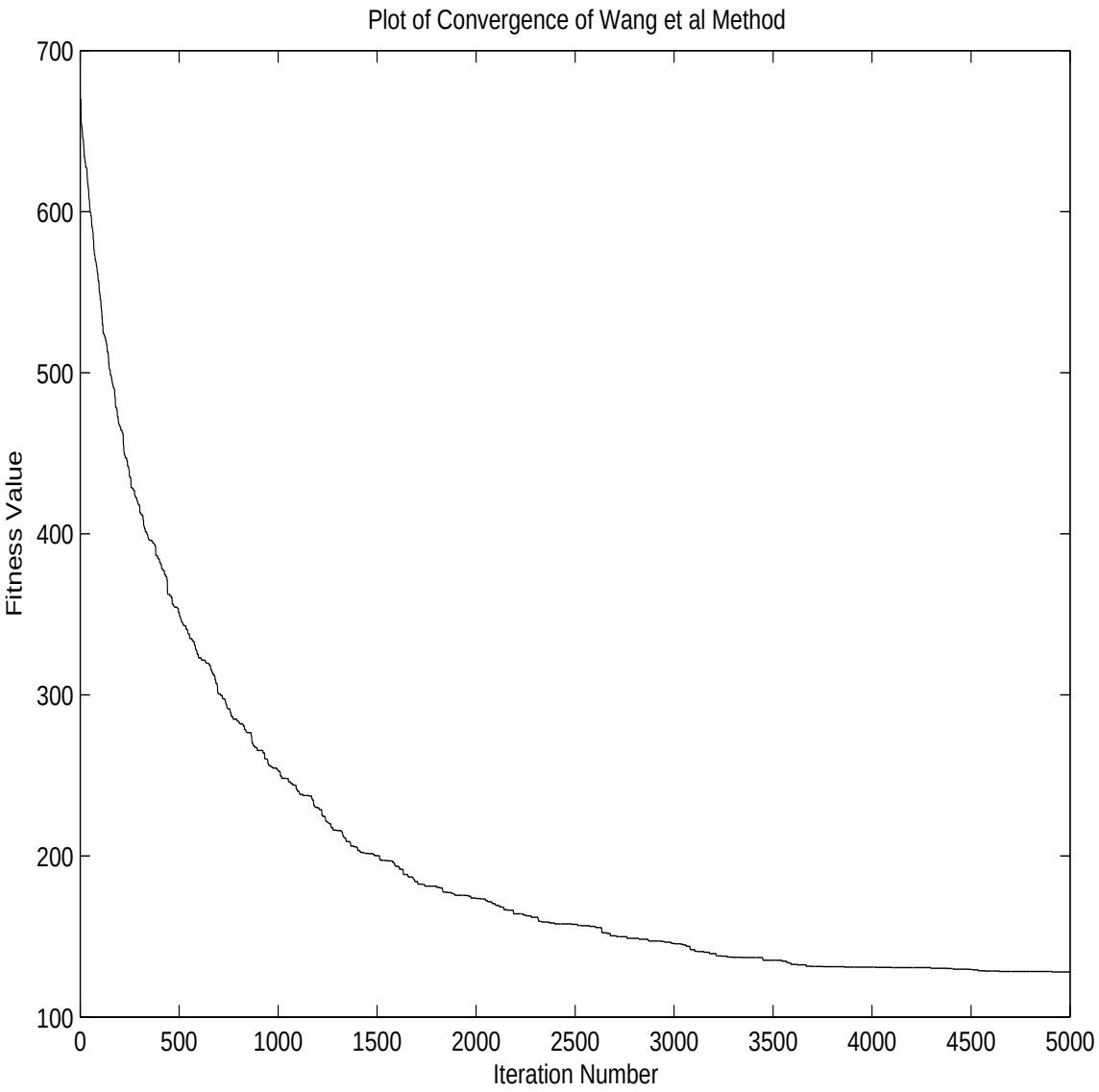

Figure 8.14: Convergence of Wang et al. Method on Simulated Data Set 8-3.

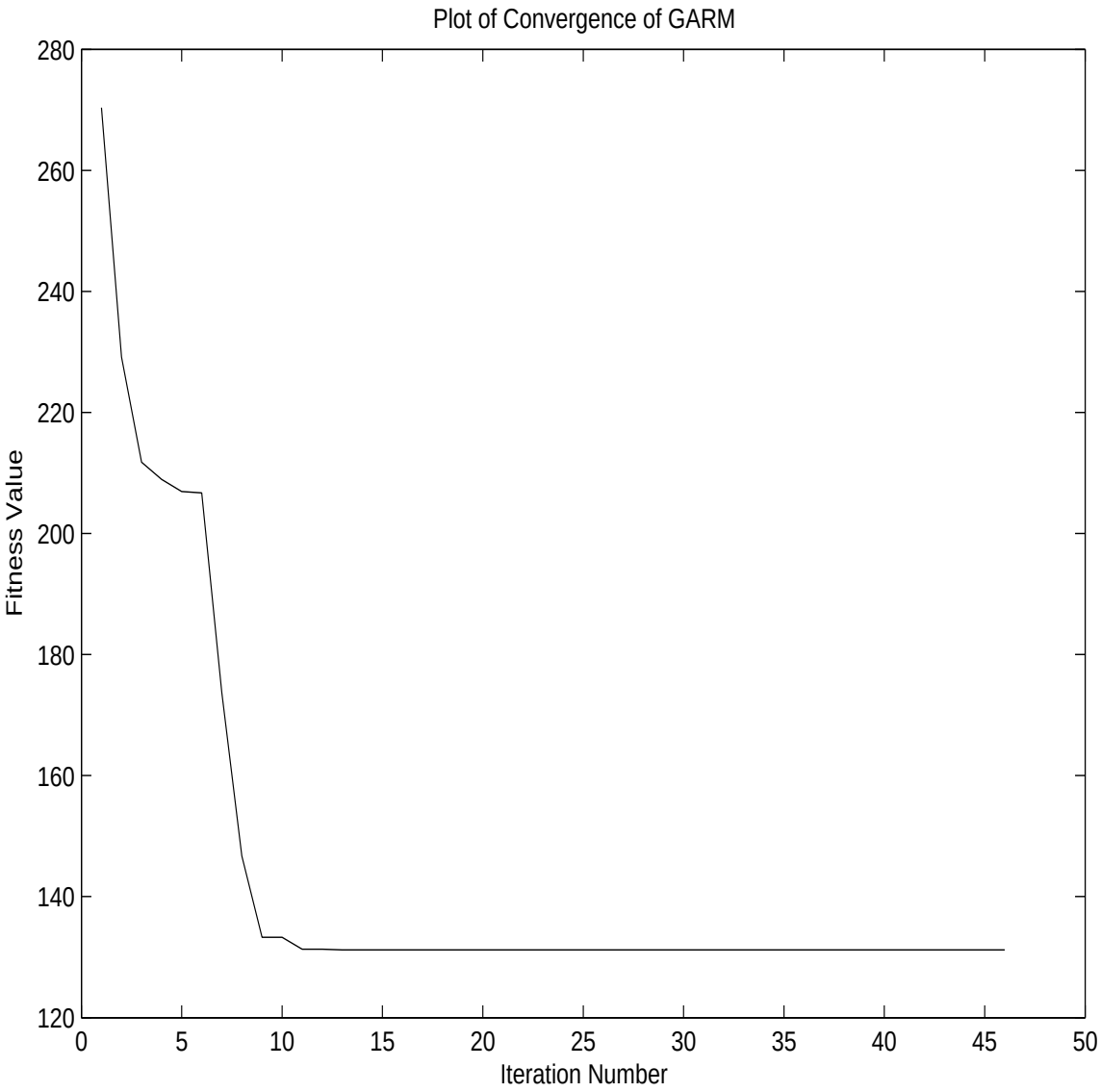

Figure 8.15: Convergence of GARM on Simulated Data Set 8-3.

structures. Numerical simulation trials tested the effectiveness of our GA clustering method. We demonstrated how this algorithm can separate more complex structures than the Genetic K-Means algorithm, and converges much faster than Mahalanobis distance-based methods using traditional GA opertations. This is especially true when the clusters overlap or show complex structures. Because of its accuracy, GARM gave better estimations of the means and covariance matrices of the identified clusters than GKM. We also demonstrated that GARM converges quickly to an optimal solution. The convergence properties of GARM are much better than the Wang et al. method. We believe that this algorithm can be implemented in data mining applications and in complex pattern recognition problems.

Chapter 9

Genetic Expectation-Maximization for Multivariate Mixture Modeling

9.1 Introduction

The problem of classifying data according to definable distributions is important in many fields. The number of ways that n data points can be divided into p partitions is $\binom{n}{p}$. Because of this large number of combinations, accurately modeling the distributions of data as finite mixtures can be an almost intractable problem. According to Bozdogan (1994), "Analysis of clusters by means of mixture distributions, called mixture model cluster analysis, has been one of the most difficult problems in statistics." Efficient and accurate methods must be developed to deal with the flood of data from ever-expanding databases.

One of the tools used by researchers to model structure in data is by using finite mixtures of distributions. The mixture modeling process estimates parameters that describe distributions of data. Mixture modeling can be applied to both univariate

and multivariate data that follow any distribution with definable parameters, including mixtures of binomial, exponential, and Poisson data. Suppose that we have n observations of dimension p so that $\{x_1, x_2, \ldots, x_n\} \in \mathbf{R}^p$. We assume that these observations follow probability density g with parameters $\hat{\theta}$, and that each observation x_i has probability π_k of belonging to the kth cluster. The sample of observations can be described as arising from the mixture of probability densities

$$f(x, \theta) = \sum_{k=1}^{K} \pi_k g(\hat{\theta}) \tag{9.1}$$

where $\pi_1, \pi_2, \ldots, \pi_K$ are the mixture proportions, with $0 \leq \pi_k \leq 1$, $k = 1, 2, \ldots, K$, and $\sum_{k=1}^{K} \pi_k = 1$.

For data that follow normal distributions if cluster k has mean μ_k and covariance matrix $\mathbf{\Sigma}_k$, then the probability density is written as

$$g_k(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k) = (2\pi)^{-\frac{p}{2}} |\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right] \tag{9.2}$$

The finite mixture of normals has the form

$$f(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Sigma}) = \sum_{k=1}^{K} \pi_k g_k(\mathbf{x}; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k) \tag{9.3}$$

From this, we can derive the log-likelihood function of the data to be

$$l(\boldsymbol{\pi}, \boldsymbol{\mu}, \mathbf{\Sigma}) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k (2\pi)^{-\frac{p}{2}} |\mathbf{\Sigma}_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)\right]\right] \tag{9.4}$$

The mixture modeling process estimates parameters describing the mixture of distributions. For a mixture of normal distributions, for example, we seek parameter estimations that describe the clustered data set. To this end, researchers have

implemented the Expectation-Maximization (EM) Algorithm. Dempster, Laird, and Rubin (1977) proposed the EM algorithm as a general parameter estimation method, while Peters and Walker (1978) applied the EM algorithm to mixture modeling. This method has been widely applied in data analysis problems. Under the EM paradigm, the researcher initializes parameter estimates, and the algorithm sequentially updates estimates of the model parameters while also iteratively updating cluster assignments according to calculated parameters. A good tutorial on the traditional EM algorithm is in Chapter 8 of the text by Martinez and Martinez (2002).

While the EM algorithm proved to be a great advance in cluster analysis, some shortcomings are inherent in the method. In order to start the EM process, the researcher must have initial estimates of parameter values. However, the search space for cluster problems is generally highly nonlinear with many local maxima. The iterative algorithm can also have a very slow convergence rate. In addition, many researchers advocate the K-means algorithm as an initialization scheme (Bozdogan 1994). The K-means algorithm assumes that data are hyperspherical, and it can only accurately calculate diagonal entries in cluster covariance matrices. Many data show more complex structure, so it can be a tedious task to find good starting values for the traditional EM algorithm using K-means. Perhaps the most serious shortcoming of the traditional EM algorithm is that it is guaranteed to only find a local maximum in the log-likelihood with the quality of the final solution dependent on the initial parameter estimates.

In order to overcome the handicaps of the traditional EM algorithm, we propose a new Expectation-Maximization algorithm for mixture models based on genetic algorithms (GA's). Our algorithm, called Genetic Expectation-Maximization (GEM), implements an efficient method to search for optimal cluster assignments as measured by the log-likelihood function. Rather than using a gradient based searching method,

our new technique relies on optimization properties of GA's to search for a global maximum in the cluster log-likelihood value. It implements the same kind of data structure as the Genetic K-means (GKM) algorithm (Krishna and Murty 1999) and the GARM algorithm, making it efficient and accurate in the cluster analysis calculations. Our calculations show that this method can accurately classify data even though different trials of the algorithm may have different initialization schemes. The global optimization properties make our algorithm less sensitive to starting values than the traditional EM method.

This chapter will introduce our new GEM algorithm in the context of mixture model cluster analysis. Section 2 will review the traditional EM algorithm for finite mixture models. Section 3 will introduce the theory behind our GA-based EM algorithm and section 4 will enumerate the new method. Section 5 will give examples of our method applied to simulated data. The chapter will conclude in section 6.

9.2 Traditional EM algorithm

The EM algorithm is currently the standard method for estimating parameters that describe distributions. This method tries to calculate parameters that maximize the log-likelihood of the mixture of distributions. These parameters are used to calculate optimal cluster assignments for a mixture of distributions, generating the Maximum Likelihood Estimate (MLE) of the data set. Under the assumption of normally distributed data, the log-likelihood function of the data with n observations and k clusters is given by

$$l(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k (2\pi)^{-\frac{p}{2}} \left| \boldsymbol{\Sigma}_k \right|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} \left(\mathbf{x}_i - \boldsymbol{\mu}_k \right)^T \boldsymbol{\Sigma}_k^{-1} \left(\mathbf{x}_i - \boldsymbol{\mu}_k \right) \right] \right] \quad (9.5)$$

Under the traditional paradigm, we compute approximations coming from the partial derivatives of the log-likelihood with respect to the cluster parameters. This process derives iterative estimations of the respective parameters and the estimated posterior probability of cluster membership $P(k|\mathbf{x}_i)$ as given by

$$\hat{P}(k|\mathbf{x}_i) = \frac{\hat{\pi}_k g_k(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{k=1}^{K} \hat{\pi}_k g_k(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)} \tag{9.6}$$

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{P}(k|\mathbf{x}_i) \tag{9.7}$$

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{n\hat{\pi}_k} \sum_{i=1}^{n} \mathbf{x}_i \hat{P}(k|\mathbf{x}_i) \tag{9.8}$$

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{n\hat{\pi}_k} \sum_{i=1}^{n} \hat{P}(k|\mathbf{x}_i) \left(\mathbf{x} - \hat{\boldsymbol{\mu}}_k\right) \left(\mathbf{x} - \hat{\boldsymbol{\mu}}_k\right)^T \tag{9.9}$$

Using the traditional method, we must first restrict attention to some part of the search space and give estimates of the starting values for the respective parameters. The algorithm then iteratively maximizes the log-likelihood and updates the respective estimates for the cluster partitions. We can summarize the traditional EM iteration loop as follows:

1. Initialize estimates of the parameters according to some partition scheme. Researchers most commonly advocate K-means as a way to find starting values for the EM algorithm.

2. Calculate the posterior probabilities for each data point $\hat{P}(k|\mathbf{x}_i)$.

3. Update the values of $\hat{\pi}_k$, $\hat{\boldsymbol{\mu}}_k$, and $\hat{\boldsymbol{\Sigma}}_k$ according to the new cluster memberships.

4. Continue this process until there is no significant increase in the log-likelihood value.

At the end of the EM loop, data points $\mathbf{x}_i$ are assigned to clusters which have the highest posterior probability such that

$$\hat{\pi}_l g_l(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_l, \hat{\boldsymbol{\Sigma}}_l) \leq \hat{\pi}_k g_k(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k) \tag{9.10}$$

for all $l \neq k$. Hence, points are assigned to clusters that have the highest posterior probability of cluster membership. The parameters π_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are likewise calculated as the MLE's of the parameters describing the cluster partitions.

While this derivative-based EM procedure has enjoyed wide applicability, it has some shortcomings. In many cases, the EM algorithm converges slowly, taking hundreds to thousands of iterations. It is also a local maximizer. Depending on the starting values, the iterative EM algorithm can return different, possibly suboptimal, estimations of cluster parameters. This becomes especially problematic with high-dimensional data that shows complex covariance structure. The parameter space in these cases is generally nonlinear with many local maxima.

Another dilemma to using the traditional EM algorithm comes from data that show hyperellisoidal covariance structure. With the traditional EM method, if the researcher uses the K-means algorithm to initialize parameters, then they may have poor initial estimates of the covariance values. The K-means algorithm can only identify hyperspherical clusters that do not overlap. At best, the K-means algorithm can only accurately estimate diagonal elements of a cluster's covariance matrix. In some hyperellisoidal data sets, initial parameter values derived from the K-means procedure are far from optimal. In these cases, it may be difficult for the traditional EM method to find good starting values for cluster estimates, making accurate cluster parameter estimations nearly impossible.

In order to overcome the handicaps of the traditional EM algorithm, we propose a new GA-based method. This method accurately and efficiently searches for a global

maximum in the log-likelihood value. Numerical trials show that our method is less sensitive to initialization, so that our method can use GKM or GARM to compute the initial cluster parameters, with little difference in the final results. These innovations allow our algorithm to find optimal parameter estimates of complex hyperellisoidal clusters.

9.3 Genetic Algorithms in Mixture Models

Since the early 1990's, researchers have used GA's in cluster analysis studies. Jones and Beltramo (1991) pioneered the *string-of-group numbers* representation, where data points are represented as strings of integers which denote their cluster assignments. The collection of strings forms a population that can undergo evolutionary operations. Krishna and Murty (1999) continued this approach by developing the GKM algorithm. Their GA loop incorporates biased mutation and the Genetic K-means operation, using these operations to search for the set of cluster assignments that minimizes the Total Within Cluster Variance. Krishna and Murty demonstrated that their method generally outperforms the traditional K-means algorithm by calculating clusters that have smaller variances.

We expanded the GKM algorithm to be applicable to hyperellisoidal clusters. Our algorithm, called Genetic Algorithm with Regularized Mahalanobis (GARM) combined the accuracy of other Hyperellisoidal clustering algorithms (Wang and Xia 1997, Wang et al. 1997) with the efficiency of GKM. Both GKM and GARM optimize fitness functions describing the quality of cluster solutions. The GKM algorithm calculates the Total Within Cluster Variation $\mathbf{W} = \sum_{k=1}^{K} \mathbf{W}_k$ where $\mathbf{W}_k = \sum_{i=1}^{n_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\mathbf{x}_i - \boldsymbol{\mu}_k)$. GARM calculates the fitness as follows: For the n_k points that belong to cluster k, $\mathbf{W}_k = \sum_{i=1}^{n_k} |\boldsymbol{\Sigma}_k|^{\frac{1}{2}} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\boldsymbol{\Sigma}_k + \epsilon \mathbf{I})^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k)$. It then figures

the fitness function of the GA to be $\mathbf{W} = \sum_{k=1}^{K} \mathbf{W}_k$, with $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ defined as the mean and covariance matrix of cluster k respectively, and ϵ is a small value that protects the inversion process from singularities. GKM and GARM incorporate genetic operations that obey Markov chain properties, causing these processes to asymptotically approach a global optimum in the fitness value. Moreover, both GKM and GARM do not rely on traditional GA operations of mutation and crossover, but rather incorporate biased mutation operations. It has been shown (Bhuyan et al. 1991, Jones and Beltramo 1991) that crossover can be computationally expensive in cluster analysis problems. We demonstrated that hyperellisoidal clustering with traditional GA operations can take hundreds to thousands of iterations to converge, while the new operations caused the algorithm to quickly converge to an optimal solution.

Our proposed algorithm implements a mixture model calculation in the same kind of GA framework as GKM and GARM. The user initializes the cluster calculations by a GA based variance minimization like GKM or GARM. Our numerical trials show that the final outcome of our algorithm shows little difference between the different initialization methods, confirming the global search ability of our method. After the minimized sum of cluster variances has converged, the algorithm starts the GA loop based on the log-likelihood values for each string. The population of strings undergo GA-type operations that seek optimum values in the cluster assignments.

GEM uses analogous GA-type operations as GKM and GARM. After the minimized variance initialization, we start a loop with the genetic operations. The fitness function of the GA is the log-likelihood for the cluster with given cluster assignments of the GA string. In the GA loop, reproductive success varies with fitness.

The first genetic-type operation is the Genetic Posterior Probability operation. This operation sequentially takes each data point and calculates the posterior probabilities of group membership for each cluster. It then assigns each data point to the cluster that has the highest posterior probability of group membership. Therefore, each string that undergoes the Genetic Posterior Probability operation has the maximized log-likelihood value given the observed means and covariance values.

The other genetic-type operation is the biased string mutation operation. In this operation, the algorithm randomly selects strings and data points to undergo the mutation operation. For the chosen data points, it then calculates the posterior probability of group membership. The relative mutation probabilities are then biased so that the probability of a data point mutating to a given cluster is proportional to its posterior probability of group membership. We found that a small mutation probability worked best for fast convergence. In our numerical trials, we used a mutation probability of 0.01.

It is possible that our biased GA operation can assign points to clusters that are far outside of the defined limits of the clusters. In our implementation, after each string undergoes any mutation operation, we force the algorithm to check for possible points assigned to a cluster that are outside of cluster's boundary. In our implementation, we used 3.5 standard deviations greater than the mean Mahalanobis distance from the cluster centroid as the outlier limit. If a point if found belonging to some cluster that is outside of this limit, then it is reassigned to the cluster with the highest probability of membership. Our numerical trials showed that this strategy helps to speed convergence of the algorithm to an optimized solution.

Our process may also generate illegal strings that are missing points from one or more clusters. This causes problems with singularities in the inversion of the cluster covariance matrix. If any strings are found to be illegal, we randomly mutate points

to form singleton clusters. This handles illegal strings in an analogous way as Krishna and Murty (1999) in the GKM algorithm. The algorithm updates all of the string mixture proportions, mean values, and variance/covariance values at each iteration. It also updates the strings affected by the respective genetic operations after each calculation.

The following lists the pseudocode for the Genetic Expectation-Maximization algorithm.

1. Initialization: The population of strings is initialized by the a GA based variance minimization. For data that show hyperspherical distributions, we used the GKM to initialize the parameter estimates and initial cluster assignments. Similarly, for hyperellisoidal data, we use GARM to calculate initial assignments. However, our trials indicated that the final results show little dependence on the initial cluster calculations.

2. Mutation: Mutation mimics random variability in a population like the random changes in genetic structure found in organisms. Our mutation operation is analogous to the GKM and GARM biased mutation operations, except that we use the posterior probability of cluster membership $P(k|\mathbf{x}_i)$. The algorithm randomly selects population strings and data points to undergo mutation. After making the selection, the probability of a point mutating into a given cluster assignment is related to the probability of that data point belonging to the cluster as computed by Bayes rule. The higher the posterior probability of cluster membership for a data point, higher the probability that it will mutate into that cluster. Let $P(k|\mathbf{x}_i)$ denote the posterior probability of data point $\mathbf{x}_i$ belonging to cluster k. The probability of mutating to the respective cluster

assignments is given by

$$p_j = \frac{c_m P_{\max} - P_j}{\sum_{i=1}^{K} (c_m P_{\max} - P_i)}$$

where c_m is a constant and $P_{\max} = \max\{P_j\}$. In our case, we set $c_m = 1$.

Our algorithm also checks for cluster outliers at the end of this operation and updates the cluster parameter estimates.

Pseudocode:

Start

Randomly select population member.

Randomly generate points $\{x_1, x_2, \ldots, x_r\}$ that undergo mutation by using *mutprob*.

For $i = 1$ to r

For $j = 1$ to K

$P_j = \frac{\pi_k g_k(\mathbf{x};\mu_k,\mathbf{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k g_k(\mathbf{x};\mu_k,\mathbf{\Sigma}_k)}$

end

$p_j = \frac{c_m P_{\max} - P_j}{\sum_{i=1}^{K} (c_m P_{\max} - P_i)}$

Generate new cluster assignments for x_i according to probabilities.

p_j.

end

Check to see if string is illegal.

Check for outlier points.

Insert mutated string into population.

Update cluster parameter values.

end

3. Posterior Probability Operation: This operation randomly selects a string from the population and then assigns each point in the string to the cluster that has the highest Posterior Probability of group membership. This operation is analogous to the K-means Operation in the GKM algorithm and the Mahalanobis operation of GARM. Because strings that undergo this Posterior Probability operation generally have among the highest log-likelihood values in the population, they propagate on to subsequent generations. Including this operation greatly speeds convergence of the algorithm to a solution. Let $P(k|\mathbf{x}_i)$ denote our posterior probability of data point x_i belonging to cluster k. We can summarize this Posterior Probability operation as follows.

   Pseudocode:

   Start

   Randomly select population member.

   For $i = 1$ to n

      for $j = 1$ to K

         $P_j = \frac{\pi_k g_k(\mathbf{x}_i;\mu_k,\mathbf{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k g_k(\mathbf{x}_i;\mu_k,\mathbf{\Sigma}_k)}$

      end

      $x_i = find(P_j = \max(P_j))$

   end

   Check to see if string is illegal.

   Insert mutated string into population.

   Update cluster parameter values.

   end

4. Selection: GEM uses the same kind of reproduction strategy as GKM and GARM. The log-likelihood value becomes the fitness of the string $F(s_i)$. The probability of reproduction then is given by

$$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}$$

Members of the next generation are selected according to their relative fitness values in a roulette wheel selection method. We include the same kind of σ-truncation method as Krisha and Murty (1999).

Pseudocode:

Start

for $i = 1$ to N

$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}$

Check σ-truncation of fitness values

end

Select members of the next generation according to probabilities $P(s_i)$.

end

9.4 Analysis

In this section, we compared the performance of our GEM algorithm on simulated data. We tested both data that show spherical symmetry and the more general case of ellipsoidal data. In the plots for these examples, different colors and symbols denote cluster membership. Black plus signs (+) mark the positions of the respective cluster means.

Table 9.1: Comparisons of the Mean Estimations of GKM and GEM for Different Clusters in Simulated Data Set 9-1.

Cluster	Original	GEM	GKM
1	(8.9,8.9)	(8.9, 8.9)	(8.8, 8.9)
2	(1.1,-0.7)	(1.1, -0.8)	(0.9, -1.3)
3	(6.1, 0.7)	(6.1, 0.8)	(5.6, 0.9)

9.4.1 Example 1

The first example of the algorithm used 3 simulated normally-distributed bivariate clusters. This data set was spherical. It had 500 data points, with 200 data points assigned to two clusters and 100 data points assigned to one cluster. The true classification of the simulated data is given in figure 9.1.

When we applied our GEM algorithm with GKM initialization to this data set, the successful classification rate was 94.8%. Figure 9.2 gives the classification results of our GARM algorithm on this data set.

As a comparison, we also classified the same simulated data set with GKM. Many researchers advocate using K-means in classification problems for data that shows spherical symmetry. Figure 9.3 shows the classification results of this trial. The classification accuracy rate for GKM was 91.0%.

We can compare the parameter estimates from GKM and GEM on this data set. The following tables compare the parameter estimates of the two methods. We can see from tables 9.1 and 9.2 that the GEM algorithm calculated better mean and covariance estimations than GKM.

Figure 9.4 shows the convergence of GEM on this data set. We can see that, starting from the GKM initialization, GEM converged to the optimum fitness value within 9 iterations.

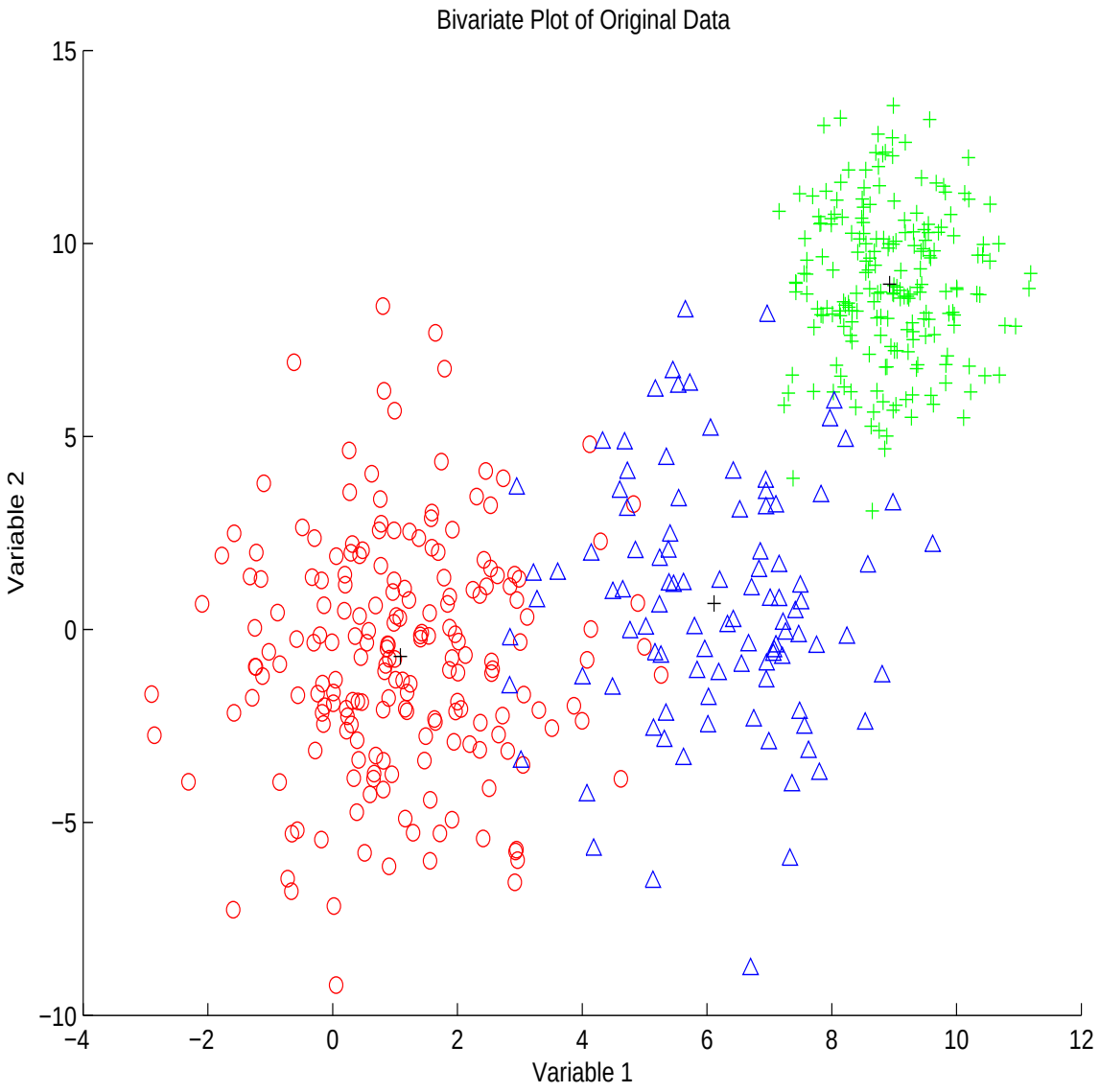

Figure 9.1: True Classification of Simulated Data Set 9-1.

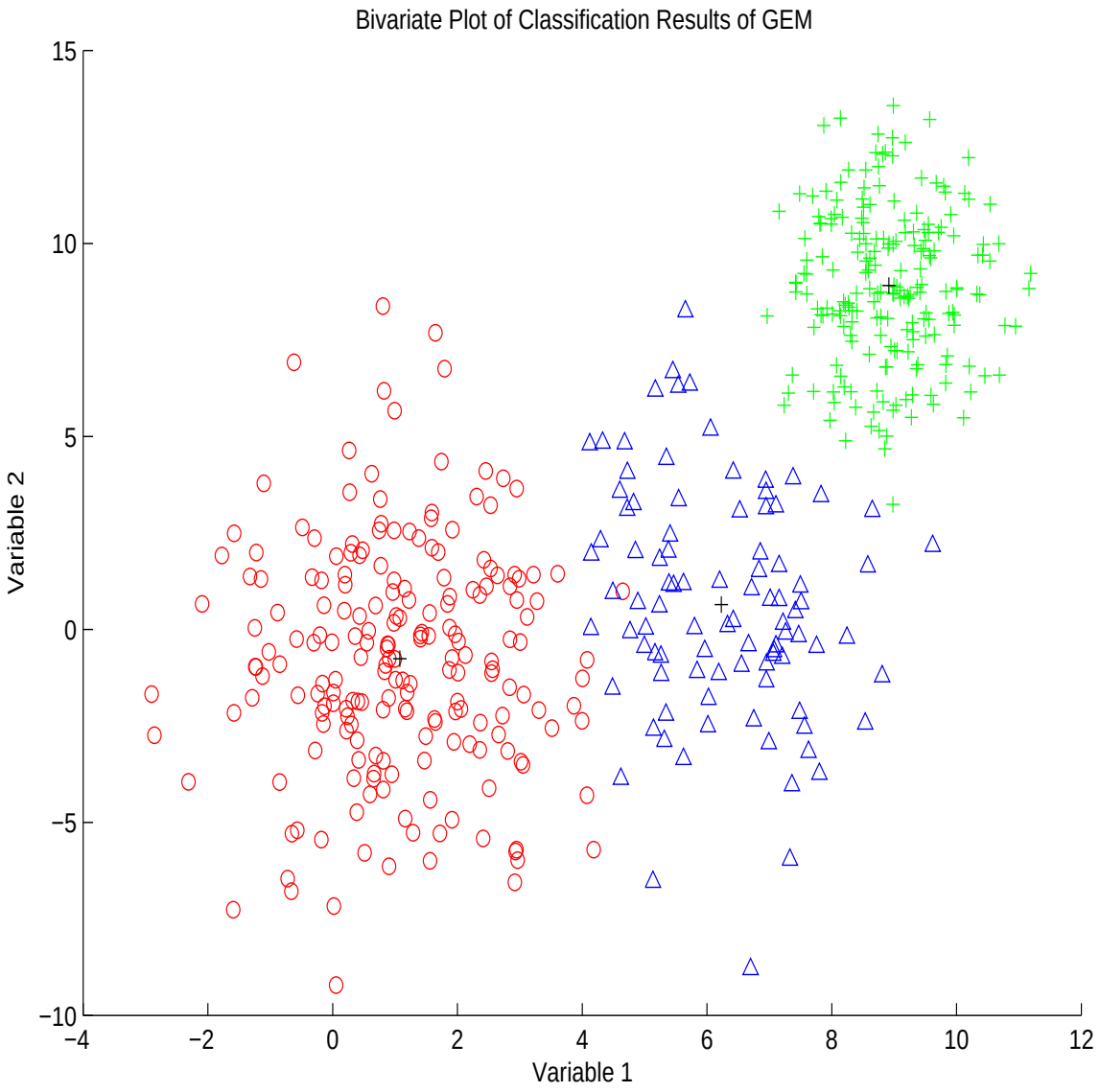

Figure 9.2: Classification Results of GEM on Simulated Data Set 9-1.

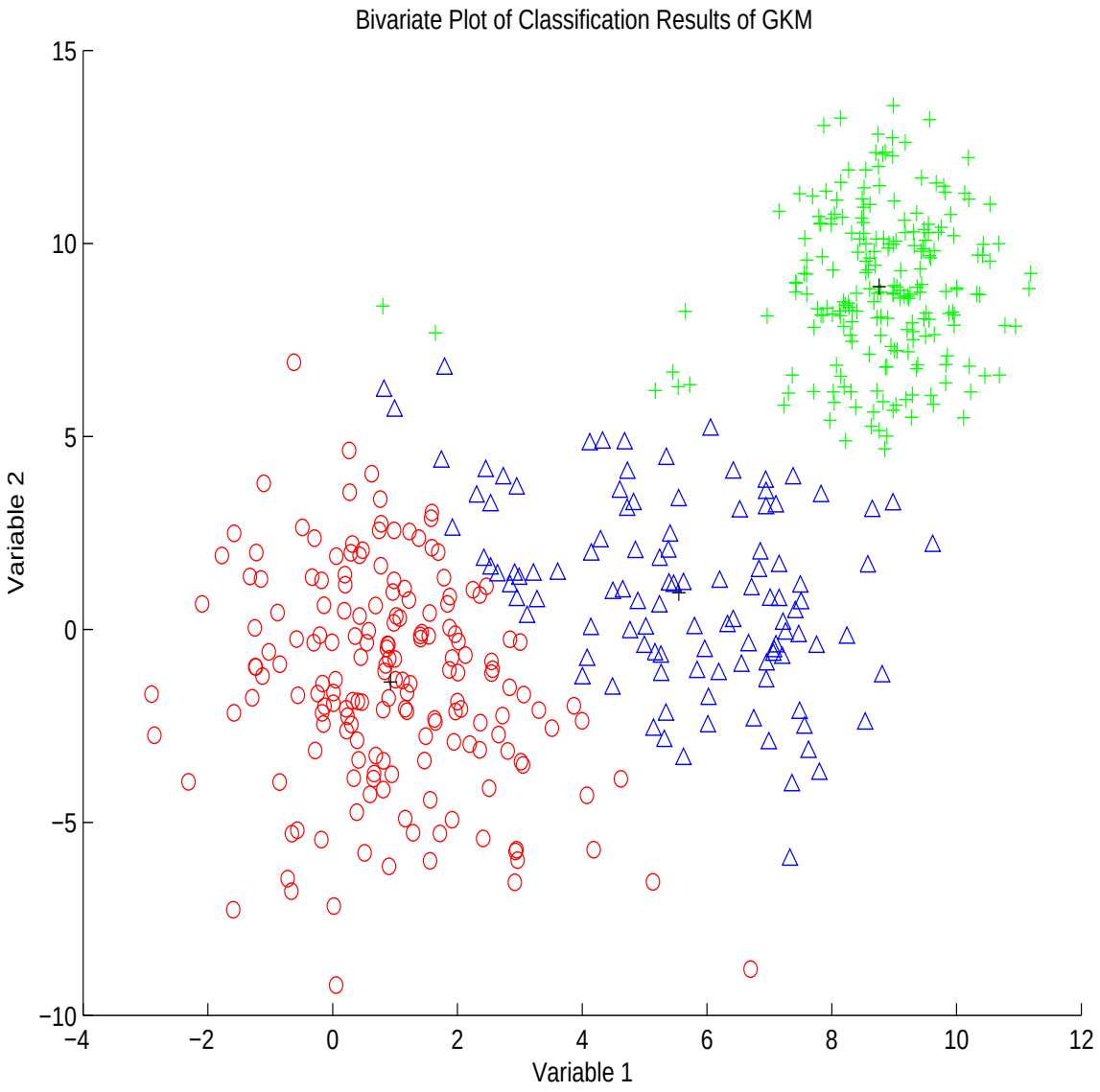

Figure 9.3: Classification Results of GKM on Simulated Data Set 9-1.

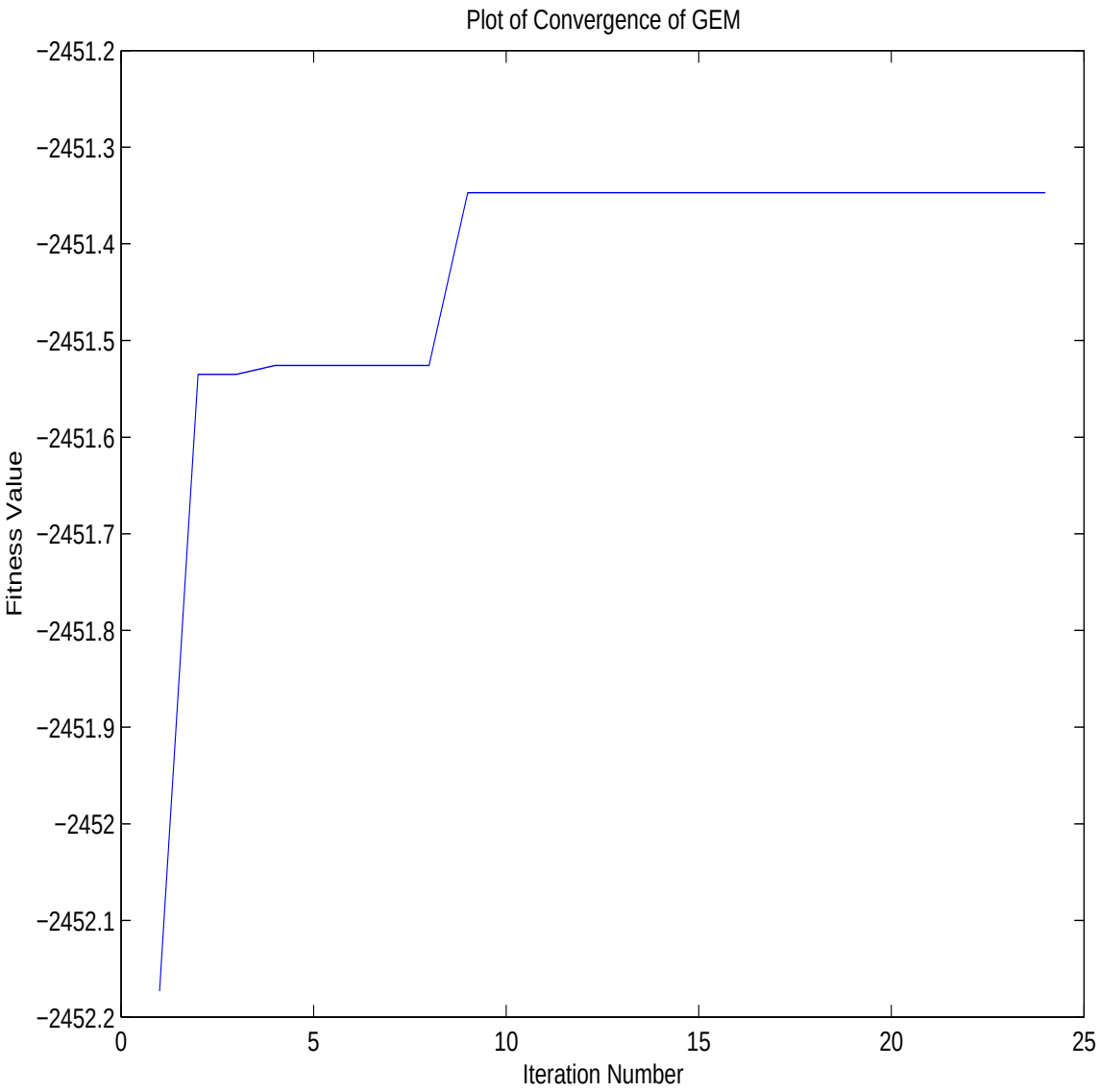

Figure 9.4: Convergence of GEM Results on Simulated Data Set 9-1.

Table 9.2: Comparisons of the Covariance Estimations of GEM and GKM for Different Clusters in Simulated Data Set 9-1.

Cluster	Original	GEM	GKM
1	$\begin{pmatrix} 0.7 & 0.0 \\ 0.0 & 4.0 \end{pmatrix}$	$\begin{pmatrix} 0.7 & 0.0 \\ 0.0 & 4.0 \end{pmatrix}$	$\begin{pmatrix} 1.5 & 0.2 \\ 0.2 & 3.8 \end{pmatrix}$
2	$\begin{pmatrix} 2.3 & 0.2 \\ 0.2 & 9.0 \end{pmatrix}$	$\begin{pmatrix} 2.1 & 0.2 \\ 0.2 & 9.0 \end{pmatrix}$	$\begin{pmatrix} 2.2 & -1.0 \\ -1.0 & 7.6 \end{pmatrix}$
3	$\begin{pmatrix} 2.2 & 0.0 \\ 0.0 & 10.0 \end{pmatrix}$	$\begin{pmatrix} 1.5 & -0.8 \\ -0.8 & 8.9 \end{pmatrix}$	$\begin{pmatrix} 3.6 & -1.9 \\ -1.9 & 5.9 \end{pmatrix}$

9.4.2 Example 2

The second example of the algorithm used 3 simulated normally-distributed bivariate clusters. This data set had 500 data points, with 200 data points assigned to two clusters and 100 data points assigned to one cluster. The distributions in this data set departed from being hyperspherical, but not strongly so. Two of the three clusters were nearly spherical, with one showing more strong ellipsoidal character. The true classification of the simulated data is given in figure 9.5.

When we applied our GEM algorithm with GARM initialization to this data set, the successful classification rate was 97.8%. Figure 9.6 gives the classification results of our GARM algorithm on this data set.

Figure 9.7 shows the convergence of GEM on this data set. Starting from the GARM initialization, GEM converged to the optimum fitness value after 23 iterations.

As a comparison, we also tried GEM with the GKM initialization. The classification accuracy of this trial was 98.2%. This classification accuracy is almost equal to that with GARM initialization. Figure 9.8 shows a plot of the classification results of this trial.

The final log-likelihood value of the GARM initialization trial was -1803.2. The final log-likelihood value of the GKM initialization trial was -1804.0. The trial with the GARM initialization calculated a slightly higher log likelihood value.

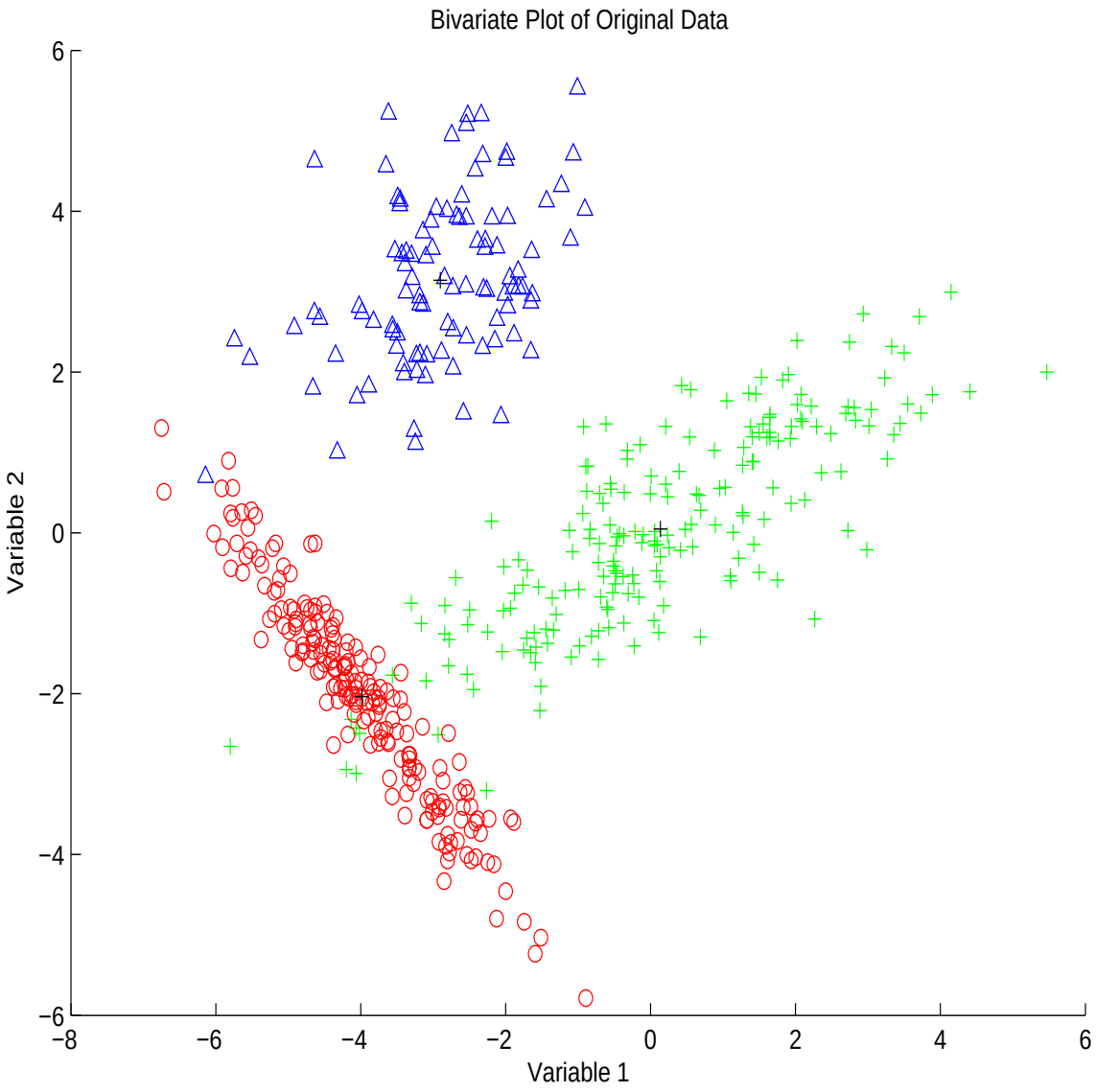

Figure 9.5: True Classification of Simulated Data Set 9-2.

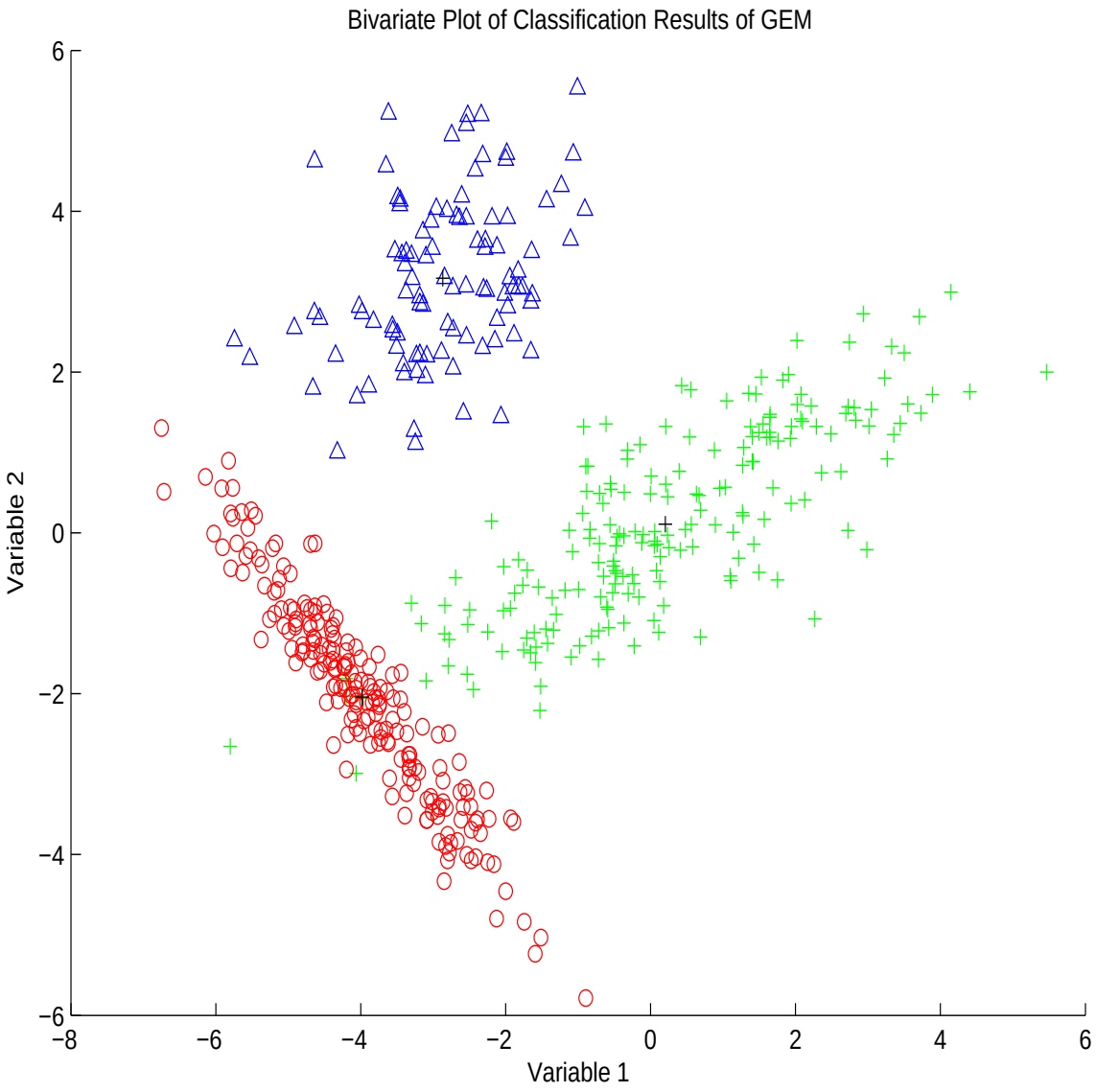

Figure 9.6: Classification Results of GEM on Simulated Data Set 9-2.

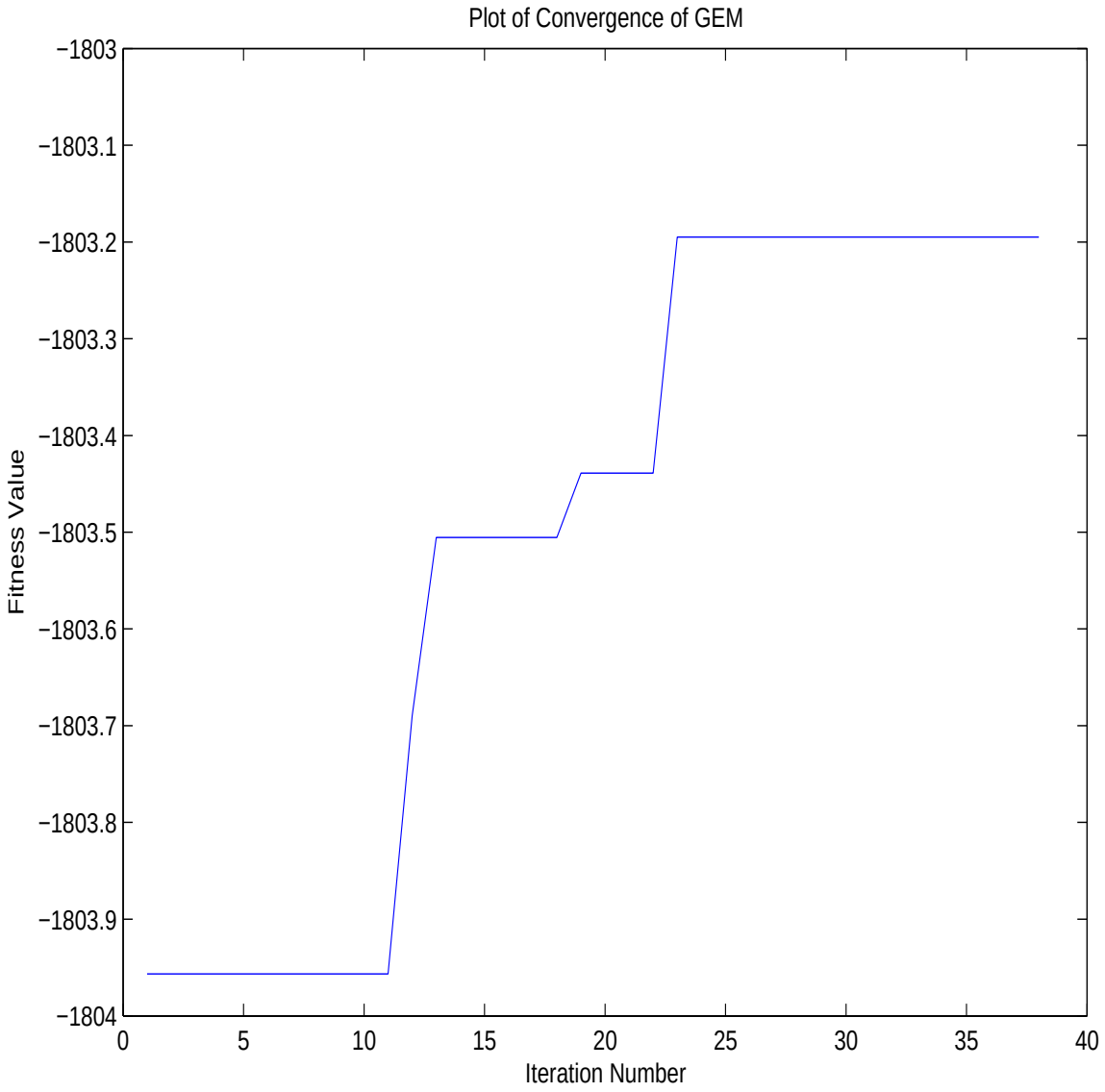

Figure 9.7: Convergence of GEM Results with GARM Initialization on Simulated Data Set 9-2.

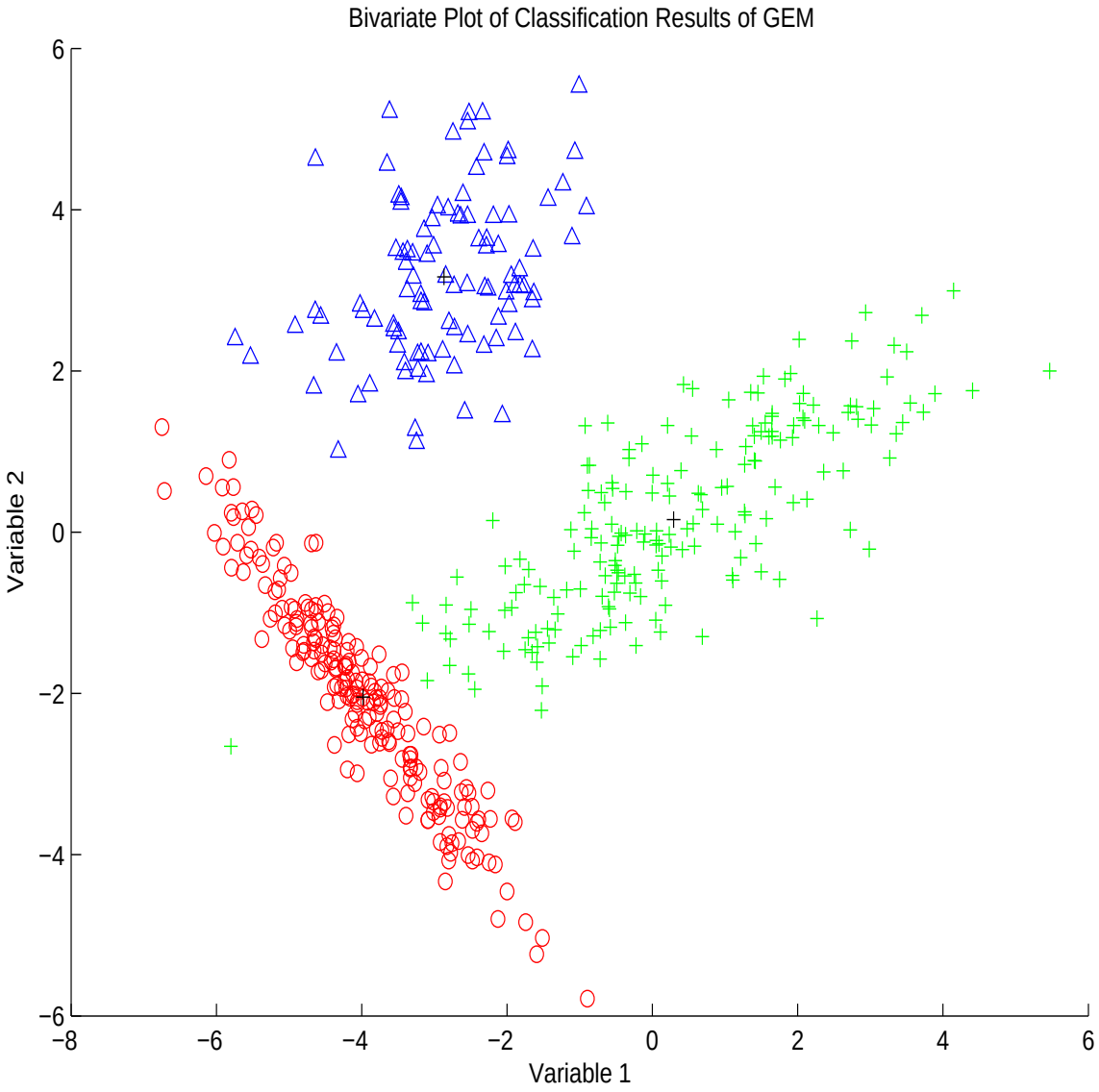

Figure 9.8: Classification Results of GEM with GKM Initialization on Simulated Data Set 9-2.

Tables 9.3 and 9.4 compare the parameter estimates of GEM with GARM initialization and GEM with GKM initialization to the true cluster parameter. The results show that GEM returned accurate parameter estimations with both initialization methods.

Figure 9.9 shows the convergence of GEM using GKM initialization on this data set. The algorithm quickly converged to an optimum in only 2 iterations after the GKM initialization.

9.4.3 Example 3

The third demonstration of GEM used 3 simulated normally-distributed bivariate clusters. This data set was ellipsoidal. This data set had 500 data points. The clusters had 150 points, 250 points, and 100 points respectively. The true classification of the simulated data is given in figure 9.10.

When we applied GEM with GARM initialization, the successful classification rate was 92.4%. Figure 9.11 gives the classification results of our GARM algorithm on this data set.

Figure 9.12 shows the convergence of GEM in this case. We can see that the algorithm converged from the GARM initialization after 120 iterations.

Table 9.3: Comparisons of the Mean Estimations of GEM with GARM Initialization and GEM with GKM Initialization for Different Clusters in Simulated Data Set 9-2.

Cluster	Original	GEM/GARM	GEM/GKM
1	(0.1,0.0)	(0.2, 0.1)	(0.3, 0.2)
2	(-4.0,-2.0)	(-4.0, -2.0)	(-4.0, -2.0)
3	(-2.9, 3.1)	(-2.9, 3.2)	(-2.9, 3.1)

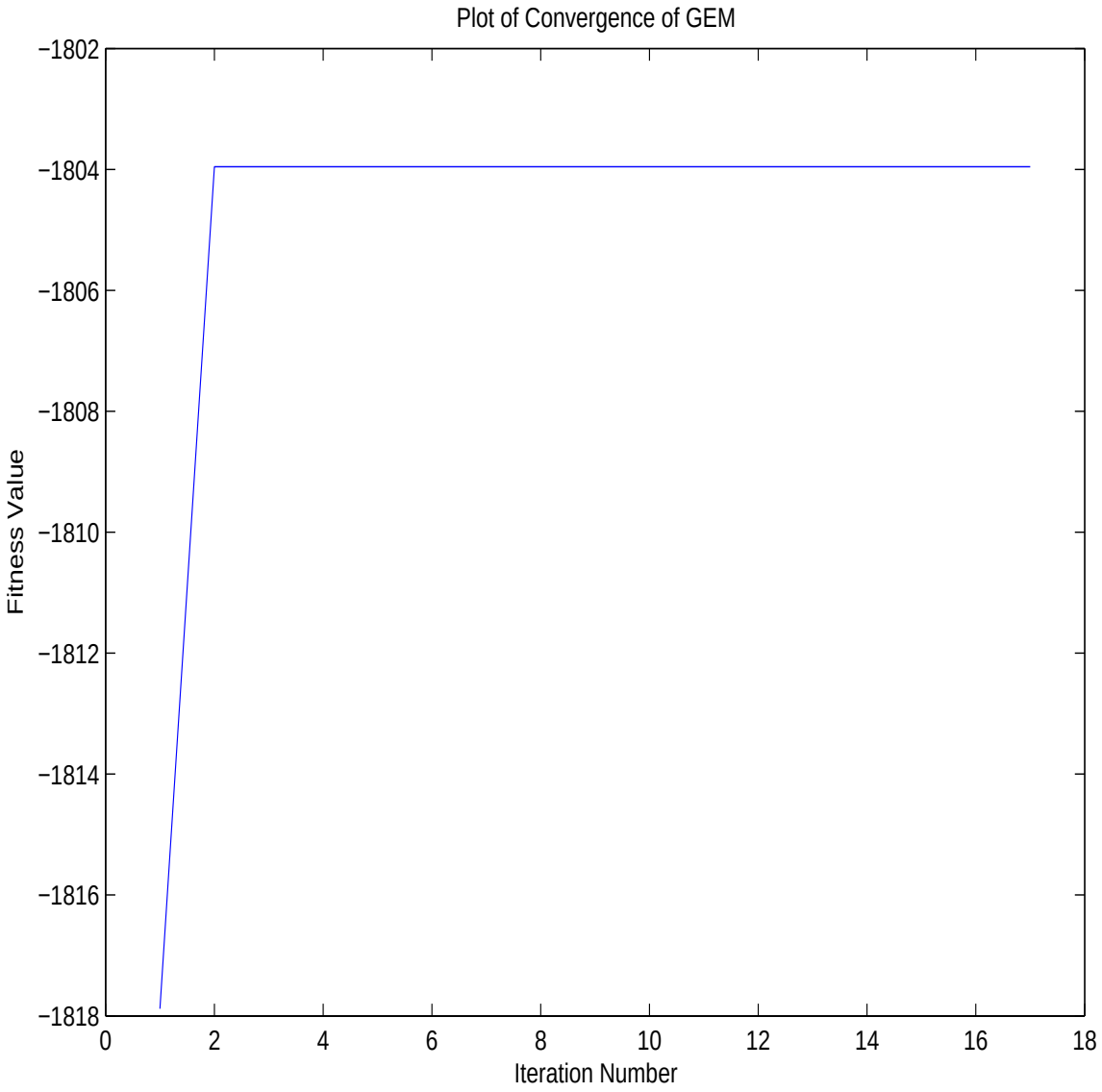

Figure 9.9: Convergence of GEM results with GKM Initialization on Simulated Data Set 9-2.

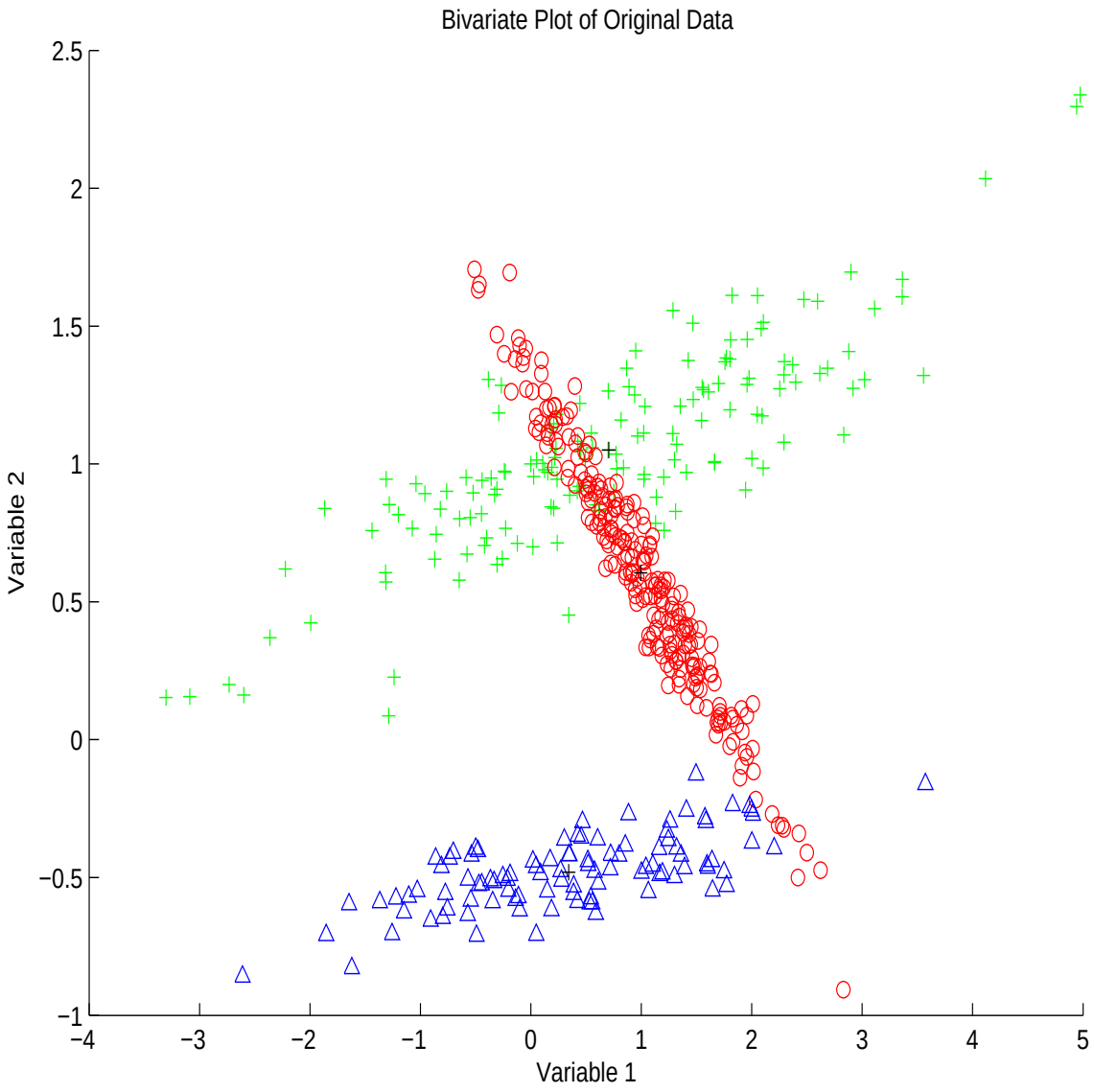

Figure 9.10: True Classification of Simulated Data Set 9-3.

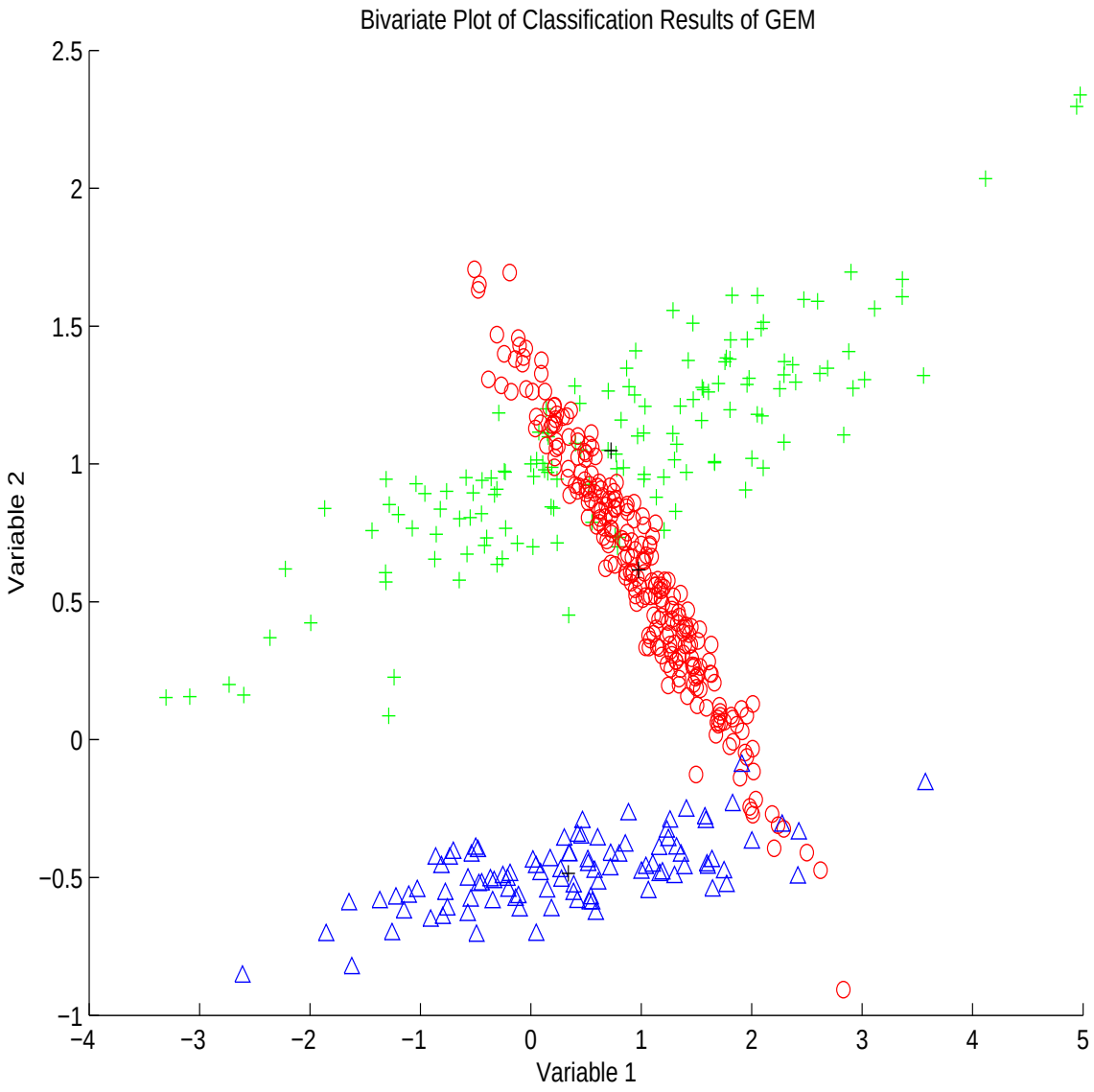

Figure 9.11: Classification Results of GEM on Simulated Data Set 9-3.

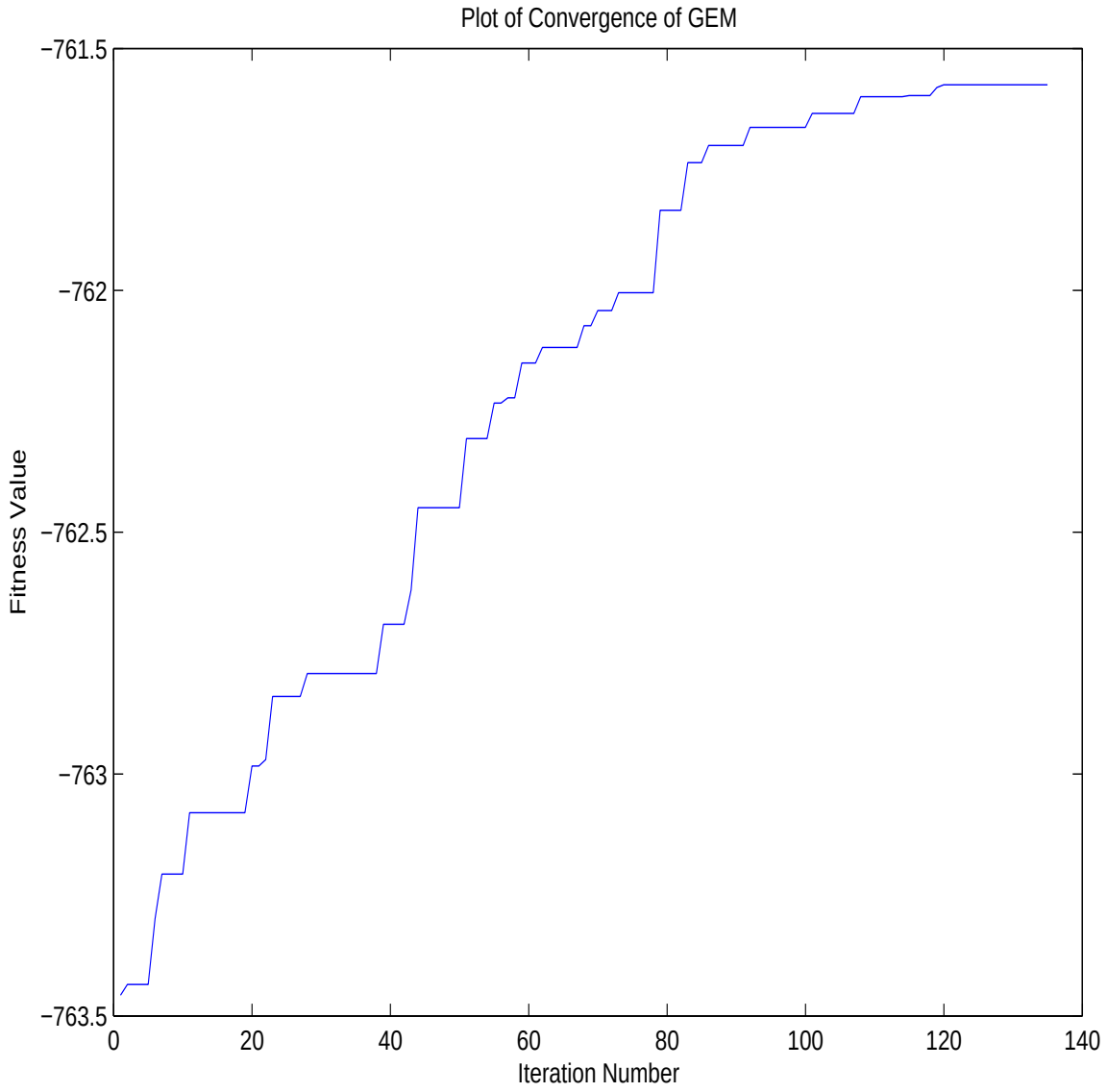

Figure 9.12: Convergence of GEM Results with GARM Initialization on Simulated Data Set 9-3.

Table 9.4: Comparisons of the Covariance Estimations of GEM with GARM Initialization and GEM with GKM Initialization for Different Clusters in Simulated Data Set 9-2.

Cluster	Original	GEM/GARM	GEM/GKM
1	$\begin{pmatrix} 3.7 & 2.0 \\ 2.0 & 1.6 \end{pmatrix}$	$\begin{pmatrix} 3.7 & 2.0 \\ 2.0 & 1.4 \end{pmatrix}$	$\begin{pmatrix} 3.3 & 1.6 \\ 1.6 & 1.3 \end{pmatrix}$
2	$\begin{pmatrix} 1.1 & -1.3 \\ -1.3 & 1.7 \end{pmatrix}$	$\begin{pmatrix} 1.1 & -1.3 \\ -1.3 & 1.7 \end{pmatrix}$	$\begin{pmatrix} 1.1 & -1.3 \\ -1.3 & 1.7 \end{pmatrix}$
3	$\begin{pmatrix} 1.0 & 0.4 \\ 0.4 & 1.1 \end{pmatrix}$	$\begin{pmatrix} 0.9 & 0.3 \\ 0.3 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 1.0 & 0.4 \\ 0.4 & 1.1 \end{pmatrix}$

As a comparison, we also applied GEM with GKM initialization to this data set. The successful classification rate in this case was also 92.4%, demonstrating the global optimization property of GEM. Figure 9.13 shows the results of this calculation.

Our calculations showed that the final log-likelihood value of the GARM initialization trial was -761.6. The final log-likelihood value of the GKM initialization trial was -762.1. Again, the trial with the GARM initialization calculated a slightly higher log-likelihood value.

We can see the parameter estimates of GEM with GARM initialization and GEM with GKM initialization in Tables 9.5 and 9.6. This case again shows that GEM returned accurate parameter estimations with both initialization methods.

Figure 9.14 shows the convergence of this trial. We can see that GEM converged to the final value after 140 iterations from the GKM initialization.

9.5 Conclusion

This chapter introduced a new GA based method for calculating mixture models and MLE's of cluster parameters. Our numerical trials showed that the new method had high rates of correct classifications and returned good cluster parameter estimations. Although our numerical trials showed little difference in the accuracies between the

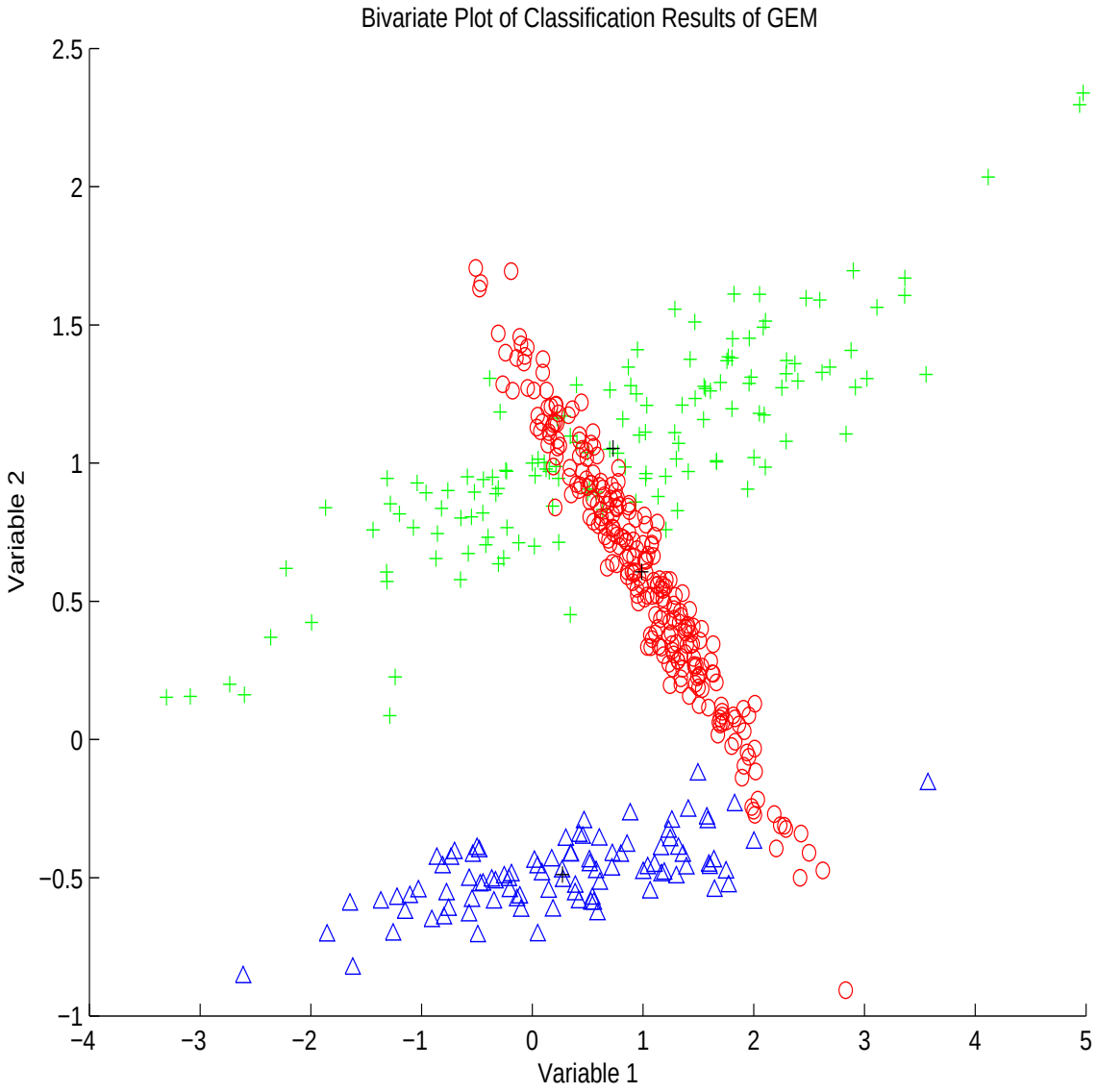

Figure 9.13: Classification Results of GEM with GKM Initialization on Simulated Data Set 9-3.

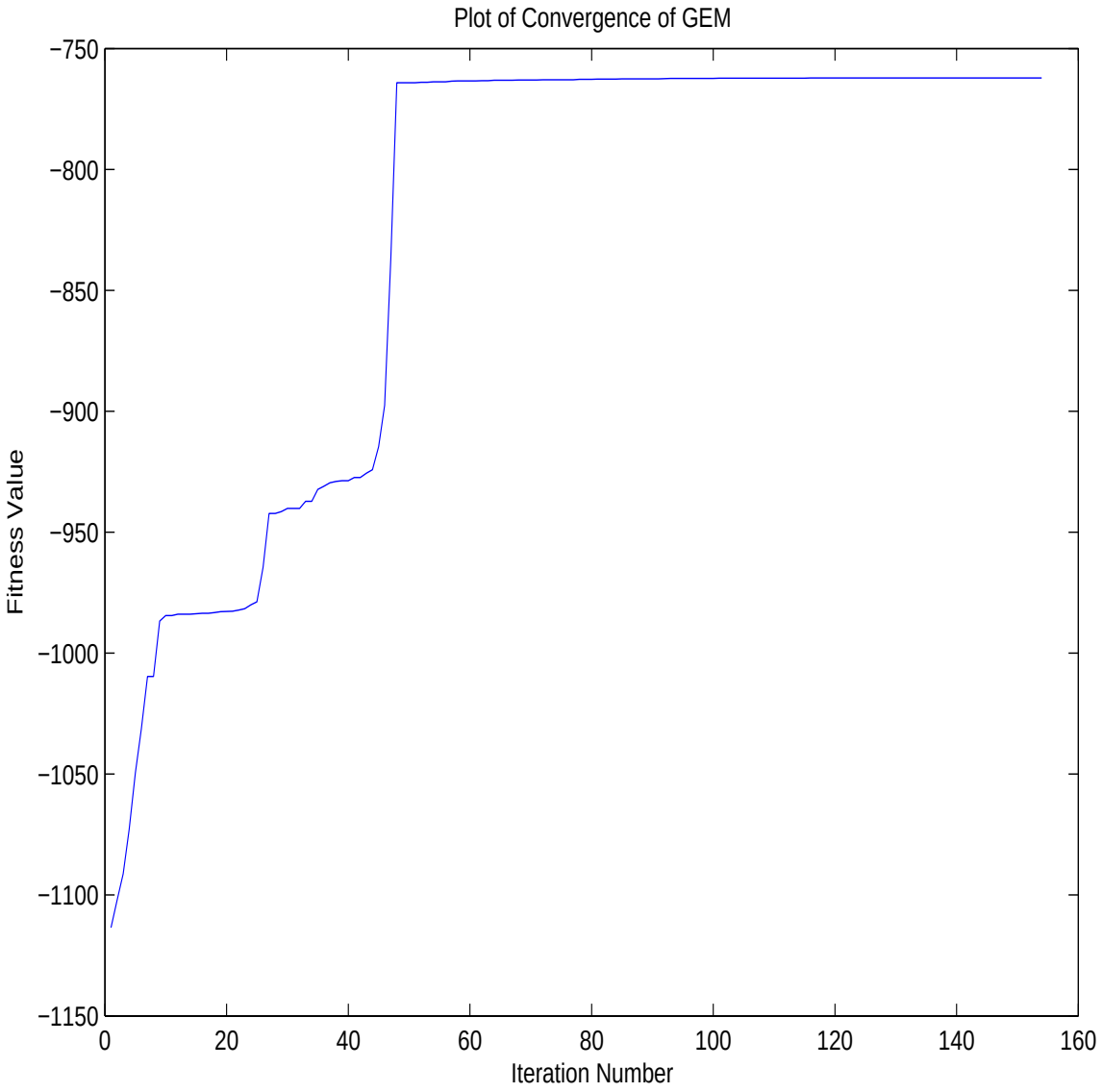

Figure 9.14: Convergence of GEM Results with GKM Initialization on Simulated Data Set 9-3.

Table 9.5: Comparisons of the Mean Estimations of GEM with GARM Initialization and GEM with GKM Initialization for Different Clusters in Simulated Data Set 9-3.

Cluster	Original	GEM/GARM	GEM/GKM
1	(0.7,1.1)	(0.7, 1.0)	(0.7, 1.1)
2	(1.0,0.6)	(1.0, 0.6)	(1.0, 0.6)
3	(0.3, -0.5)	(0.3, -0.5)	(0.3, -0.5)

Table 9.6: Comparisons of the Covariance Estimations of GEM with GARM Initialization and GEM with GKM Initialization for Different clusters in Simulated Data Set 9-3.

Cluster	Original	GEM/GARM	GEM/GKM
1	$\begin{pmatrix} 2.3 & 0.5 \\ 0.5 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 2.3 & 0.5 \\ 0.5 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 2.4 & 0.5 \\ 0.5 & 0.1 \end{pmatrix}$
2	$\begin{pmatrix} 0.4 & -0.3 \\ -0.3 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.4 & -0.3 \\ -0.3 & 0.2 \end{pmatrix}$	$\begin{pmatrix} 0.4 & -0.3 \\ -0.3 & 0.2 \end{pmatrix}$
3	$\begin{pmatrix} 1.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 1.1 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$	$\begin{pmatrix} 1.0 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$

GKM and GARM initialization, the GARM initialization generally calculated higher log-likelihood values. This difference may become more significant in analysis cases that have a larger number of clusters. We believe that this new algorithm can be used in many multivariate classification studies and that is can be implemented as a tool for data mining.

Chapter 10

Cluster Analysis with Information Scores

10.1 Introduction

Cluster analysis tries to categorize unknown data into a number of definable classes. Modern data analysis methods must find ways of processing increasingly voluminous data sets. The issue of finding the best number of clusters is an important one with applications in many fields. Initially, the researcher only has values from some experiment or observational study. Cluster analysis tries to partition the observed data according to some metric or property in the data, or tries to find some structure in the data. Not only does cluster analysis calculate parameters that describe the data, but it can also reveal structure in the data that may not be apparent otherwise.

Mixture modeling has been a favorite method for estimating parameters that describe distributions of data. Mixture models regard data as arising from distributions with definable parameters. Each data point has some posterior probability of belonging to each cluster, defined by Bayes rule. For example, with normally distributed

data with cluster means $\boldsymbol{\mu}_k$, cluster covariances $\boldsymbol{\Sigma}_k$, and mixture proportions π_k, the posterior probability of cluster membership for each data point $\mathbf{x}_i$ is

$$P(k|\mathbf{x}_i) = \frac{\pi_k g_k(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} \pi_k g_k(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \tag{10.1}$$

An added complication arises when the researcher tries to find the best number of clusters in a given data set. The problem of simultaneously deciding the best number of cluster assignments and the best number of clusters has been addressed by many authors including Beale (1969), Marriot (1971), Calinski and Harabasz (1969), Maronna and Jacovkis (1974), Hartigan (1975), Matusita and Ohsumi (1980) and Bozdogan (1994). While many methods simply give empirical guidelines, Akaike (1973) introduced information scoring as a way to remove subjectivity in statistical analysis. Bozdogan (1994) proposed implementing information scoring as an objective method for calculating the best number of clusters in a multivariate data set. Information scoring functions use penalized maximum likelihood estimation techniques. Improved methods of classification and cluster analysis must be found to handel ever-growing data sets. In addition to accurately classifying the data, such methods must be able to accurately uncover the covariance structure of the data and estimate the best number of component clusters.

We propose implementing the Genetic Expectation Maximization (GEM) algorithm into a framework that can identify the best number of clusters in a multivariate data set using measures of information complexity. GEM uses the efficient searching properties of genetic algorithms (GA's) to calculate maximum likelihood estimates of a data set. Because of its accuracy, GEM can return optimal estimates of cluster parameters. Its fast convergence also makes it ideal for complex clustering problems. By combining GEM with information based scoring, we can have an accurate and efficient tool for analyzing complex multivariate data. Section 2 of this chapter will

review the EM clustering process, while section 3 will show the development of information based methods to decide the best number of clusters. Section 4 will then propose a new way of integrating GA based clustering methodology with information complexity scoring. The advantage of this approach to that the accuracy and efficiency of GA based clustering is combined with the discriminative power of information scoring. Section 5 will show examples of our new clustering method on simulated data. The chapter will conclude in section 6.

10.2 Mixture Model Cluster Analysis

Mixture modeling tries to assign the highest probability of membership for all data points in the set, yielding the Maximum Likelihood Estimate (MLE) of the data. The EM algorithm (Dempster et al. 1977, Peters and Walker 1978) has historically provided a framework for cluster parameter estimation. For normally distributed data, parameters that describe cluster k are the mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. The mixture modeling process regards each observation as having probability π_k of coming from class k, with $k \in \{1, 2, \ldots, K\}$. Observation vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ form a sample of the mixture

$$f(\mathbf{x}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \pi_k g_k \left(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{10.2}$$

Here, $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$ are the mixture proportions such that

$$0 \leq \pi_k \leq 1 \text{ for } k = 1, 2, \ldots, K \text{ and } \pi_K = 1 - \sum_{k=1}^{K-1} \pi_k \tag{10.3}$$

Here, $g_k(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the multivariate normal density function

$$g_k(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = (2\pi)^{-\frac{p}{2}} |\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\right] \tag{10.4}$$

Under the traditional EM paradigm, the data set is first partitioned according to some clustering process, such as K-means (Bozdogan 1994). The initial cluster partitions give the starting values for cluster parameter estimates. These parameter estimates and posterior probabilities are then iteratively recalculated by a gradient ascent type process until convergence occurs.

While the traditional EM algorithm has enjoyed wide application in many mixture modeling scenarios, it has some shortcomings. The traditional EM algorithm is guaranteed only to be a local maximizer. The iterative nature of the EM algorithm can have a slow rate of convergence. Given initial estimates of distribution parameters, the EM algorithm converges to a local maximum in the parameter values. However, cluster parameter space is generally highly nonlinear with many local optima. It can be very difficult for the traditional EM algorithm to find a global maximum due to the complex parameter structure.

To overcome these limitations, we proposed a mixture modeling process based on GA strings. Our algorithm, called Genetic Expectation Maximization (GEM), uses the global search properties of GA's to maximize the log-likelihood of the data set. We demonstrated that our GEM algorithm can accurately model multivariate data with complex covariance structure.

Because it can accurately model the covariance structure of multivariate data, we propose using GEM in an information scoring cluster analysis algorithm. Accurately modeling cluster covariance structure becomes especially important when the data is overlapped or seriously departs from being hyperspherical. In order to best use

information scoring functions to identify the best number of clusters in a data set, the algorithm must accurately model the covariance structure of the data.

10.3 Information Scoring in Cluster Analysis

Deciding the number of clusters in a multivariate data set has historically been a challenging endeavor. With data having three dimensions or less, plots of the data set may reveal structure from visual inspection. There are also some published "rules of thumb" that provides guidance for deciding the best number of cluster. For example, some researchers advocate using the property that the number of data points n must be greater than the number of parameters p (Henna 1985, 1986) so that

$$K < \frac{2n}{(p+1)(p+2)} \tag{10.5}$$

Another rule of thumb says $K \approx \left(\frac{n}{2}\right)^{1/2}$ (Mardia et al., 1979). Wong (1982) suggests using $K \approx n^{0.3}$ based on empirical evidence. These guidelines involve subjectivity of the researcher in judging the number of clusters.

Since 1973, information scoring in statistics has allowed researchers to reduce or eliminate the guesswork involved in statistical methodology (Akaike 1973, 1974, 1978, 1981, 1985). The first information scoring function, Akaike Information Criterium (AIC), defines the measure of quality of a model as

$$AIC = -2 \log L\left(\hat{\Theta}\right) + 2m \tag{10.6}$$

where $logL\left(\hat{\Theta}\right)$ is the maximized log-likelihood term and m is the number of free parameters. Later, Bozdogan (1987) modified AIC into CAIC

$$CAIC = -2\log L\left(\hat{\Theta}\right) + m\left(\log(n) + 1\right) \tag{10.7}$$

In order to use these measures of complexity in determining the number of clusters, we must analyze both the maximized log-likelihood term and the number of free parameters in the cluster analysis framework. Bozdogan (1981, 1983) shows that the number of free parameters depends on the structure of the different covariance cases. The most general case is where each cluster has a different covariance matrix, and the covariance matrices have off-diagonal elements. The number of free parameters in this case is

$$m = kp + (k-1) + kp\left(p+1\right)/2 \tag{10.8}$$

where k is the number of clusters and p is the number of variables. Using this expression, we can combine the parameter counts for the covariance structure with the maximized log-likelihood values to give an expression for AIC (Bozdogan 1994):

$$AIC = -2\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}\hat{\pi}_k g_k\left(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right)\right] + 3\left[kp + (k-1) + kp\left(p+1\right)/2\right] \tag{10.9}$$

We can also derive an expressions for CAIC in the same way. For the general cluster covariance case, we have (Bozdogan 1994):

$$CAIC = -2\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}\hat{\pi}_k g_k\left(\mathbf{x}_i; \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k\right)\right] + m\left[\log(n) + 1\right] \tag{10.10}$$

A more modern measure of information complexity was developed by Bozdogan (1988, 1990a, 1990b) that more accurately discriminates the best model for a given

data set. Called ICOMP, this measure of information complexity can be written as:

$$ICOMP = -2\log L\left(\hat{\Theta}\right) + 2C_1\left(\hat{F}^{-1}\right) \tag{10.11}$$

where $\log L\left(\hat{\Theta}\right)$ is the maximized log-likelihood function and $C_1\left(\hat{F}^{-1}\right)$ is the complexity of the Fisher information matrix, defined as

$$C_1\left(\hat{F}^{-1}\right) = \frac{s}{2}\log\left(trace\left(\hat{F}^{-1}\right)/s\right) - \frac{1}{2}\log\left(\det\left(\hat{F}^{-1}\right)\right) \tag{10.12}$$

where $s = \dim(\hat{F}^{-1}) = rank(\hat{F}^{-1})$.

Bozdogan (1994) showed that ICOMP has several advantages over other information criteria. ICOMP controls the risk of both underfitting and overfitting models. It data adaptively finds a balance between lack of fit and model complexity. In addition, ICOMP automatically takes account of sample size. The model with the minimum ICOMP score is the best among all competing models for a given data set. Bozdogan (1994) derived ICOMP in a form that is useful for scoring cluster combinations under the different types of covariance structures. The basis of these ICOMP expressions is the complexity of the Fisher information matrix for cluster analysis (Bozdogan 1994):

$$\begin{aligned} C_1^*\left(\hat{F}^{-1}\right) &= (kp + kp(p+1)/2)\log[\mathbf{M}] \\ &\quad - \left\{(p+2)\sum_{k=1}^{K}\log(\det(\hat{\mathbf{\Sigma}}_k)) - \sum_{k=1}^{K}\log\left(\hat{\pi}_k n\right)\right\} - (kp)\log(2n) \end{aligned} \tag{10.13}$$

where

$$\mathbf{M} = \frac{\sum_{k=1}^{K}\left\{\frac{1}{\hat{\pi}_k}trace\left(\hat{\mathbf{\Sigma}}_k\right) + \frac{1}{2}trace\left(\hat{\mathbf{\Sigma}}_k^2\right) + \frac{1}{2}trace\left(\hat{\Sigma}_k\right)^2 + \sum_{j=1}^{P}\left(\hat{\sigma}_{kjj}\right)^2\right\}}{kp + kp\left(p+1\right)/2} \tag{10.14}$$

Inserting this expression into the definition of ICOMP yields the scoring function for the most general covariance structure:

$$ICOMP = -2\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}\hat{\pi}_k g_k\left(\mathbf{x}_i;\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}}_k\right)\right] + C_1^*\left(\hat{F}^{-1}\right) \tag{10.15}$$

Expressions of ICOMP for more specific covariance structures can be found in Bozdogan (1994).

While the preceeding expressions for information based scoring functions are true, but they are only useful if we can find a way to partition a data set while accurately estimating the means and covariances of the resulting clusters. The next section addresses this problem of simultaneously calculating cluster assignments and using the resulting means and covariances to score the number of clusters.

10.4 Information Scoring Combined with GA Clustering

In order to use these measures of information complexity, the researcher must have a way to estimate the means and covariances for a given number of clusters. They then use these estimates to calculate the score functions for different numbers of clusters. The number of clusters that achieves the minimum score value is the best in terms of fitting a normally-distributed mixture model to the observed data set.

We propose a new method of cluster analysis that can accurately and efficiently give the best number of clusters for a given data set. This new method combines the GEM Mixture Model cluster assignments with information based scoring. Previous methods of using information based scoring have used traditional seed-based

clustering methods, like K-means, to calculate the cluster assignments. The traditional seed-based methods depend on initial cluster assignments and generally return suboptimal cluster partitions of the data. Because of its suboptimal properties, the traditional K-means algorithm can return poor estimates of the clusters' means and covariances, as well as poor estimates of the log-likelihood sums used in the scoring functions. By contrast, GEM does not rely on seeds or initial assignments and generally returns more accurate cluster partitions. Consequently, the GEM cluster assignments generally give more accurate means and covariance estimates.

We propose a new algorithm that combines the accuracy and efficiency of GEM with the discriminative ability of information scoring. The algorithm can be stated as follows:

1. Start a loop that covers the desired range of cluster tests. For example, if the researcher believes that the best number of clusters is between $n_{\min} = 10$ and $n_{\max} = 20$, these become the limits of the scoring tests.

2. For each number of clusters n in the test loop, use GEM to calculate the mixture model components.

3. Assign a score to the clustering by using an information based measure like AIC or ICOMP. ICOMP is considered the most accurate measure of information complexity. In cases where some clusters may have fewer points than the number of variables, a covariance estimator can be used to control possible singularities.

4. After the end of the scoring loop, the number of clusters n that achieves the minimum information score is the best to describe the data set under study.

We will next demonstrate examples of using this method on simulated data.

Table 10.1: ICOMP and AIC Scores for the Mixture Models in Simulated Data Set 10-1.

Number of Clusters	ICOMP Score	AIC Score
1	5435.7	5435.6
2	4988.8	4998.8
3	4918.1	4936.1
4	4935.6	4967.6
5	4932.9	4978.4
6	4933.2	4987.9
7	4948.2	5011.8
8	4936.4	4996.0
9	4952.2	5017.9
10	4966.9	5004.4

10.5 Analysis

This section will demonstrate the new cluster analysis method on simulated data. The data show increasingly complex covariance structure, which tests how accurately the new cluster analysis method can model the cluster structure. A discussion of the results accompanies the numerical results.

10.5.1 Example 1

This trial used 500 simulated data points in 3 clusters. Two of the clusters contain 200 data points, while one contains 100. Figure 10.1 shows the three respective clusters which are denoted by different symbols and colors.

We applied the the information scoring algorithm to this data set. We used the GEM algorithm with GKM initialization. A listing of the information scores for the respective number of clusters is given in table 10.1.

The respective information scores for the different numbers of clusters indicate that the minimum of both AIC and ICOMP occurs at $n = 3$. This result confirms that this information based scoring method accurately identified the correct number

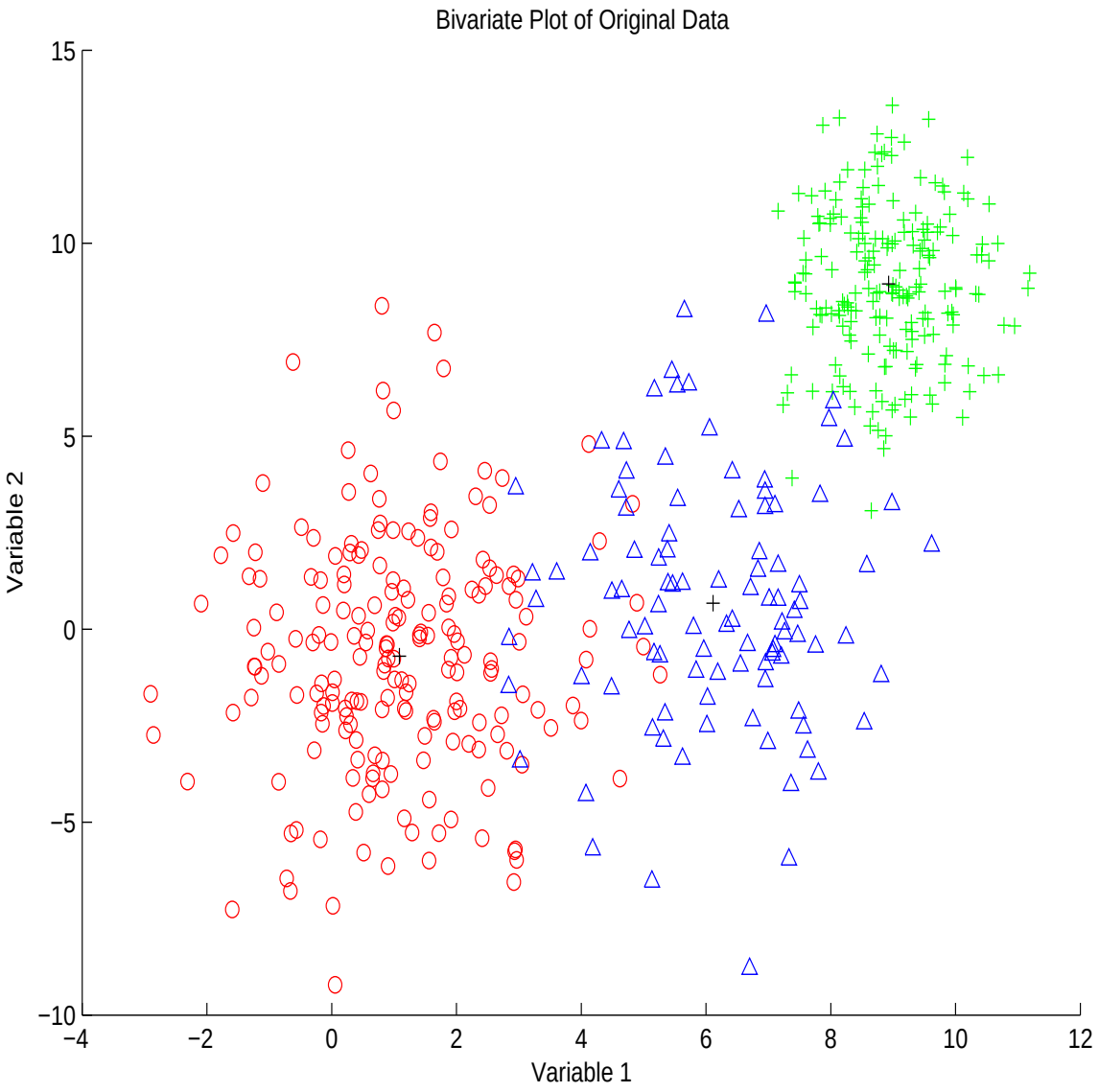

Figure 10.1: True Classification of Simulated Data Set 10-1.

Table 10.2: ICOMP and AIC Scores for the Mixture Models in Simulated Data Set 10-2.

Number of Clusters	ICOMP Score	AIC Score
1	2932.4	2939.4
2	2585.3	2591.2
3	2173.9	2173.8
4	1901.5	1900.9
5	1798.3	1802.8
6	1906.9	1794.6
7	1835.3	1819.6
8	1840.0	1819.6
9	1855.0	1822.5
10	1874.8	1844.4

of component clusters. The algorithm was able to do this automatically with no seeds or prior knowledge about the clusters. Figures 10.2 and 10.3 show plots of the ICOMP and AIC scores respectively.

10.5.2 Example 2

This trial used 500 simulated data points in 5 clusters. Each cluster had 100 points, and the clusters showed ellipsoidal structure. Figure 10.4 shows the clusters denoted by different symbols and colors.

We applied the information scoring algorithm to this data set. We used the GEM algorithm with GARM initialization. A listing of the information scores for the respective number of clusters is given in table 10.2.

The scoring results showed that the minimum value of ICOMP occurred at 5 clusters, but the minimum AIC score appeared at 6 clusters. ICOMP was able to discriminate the correct number of clusters. This example confirms how the GA based clustering can model the covariance structure of the data set, and the complexity term

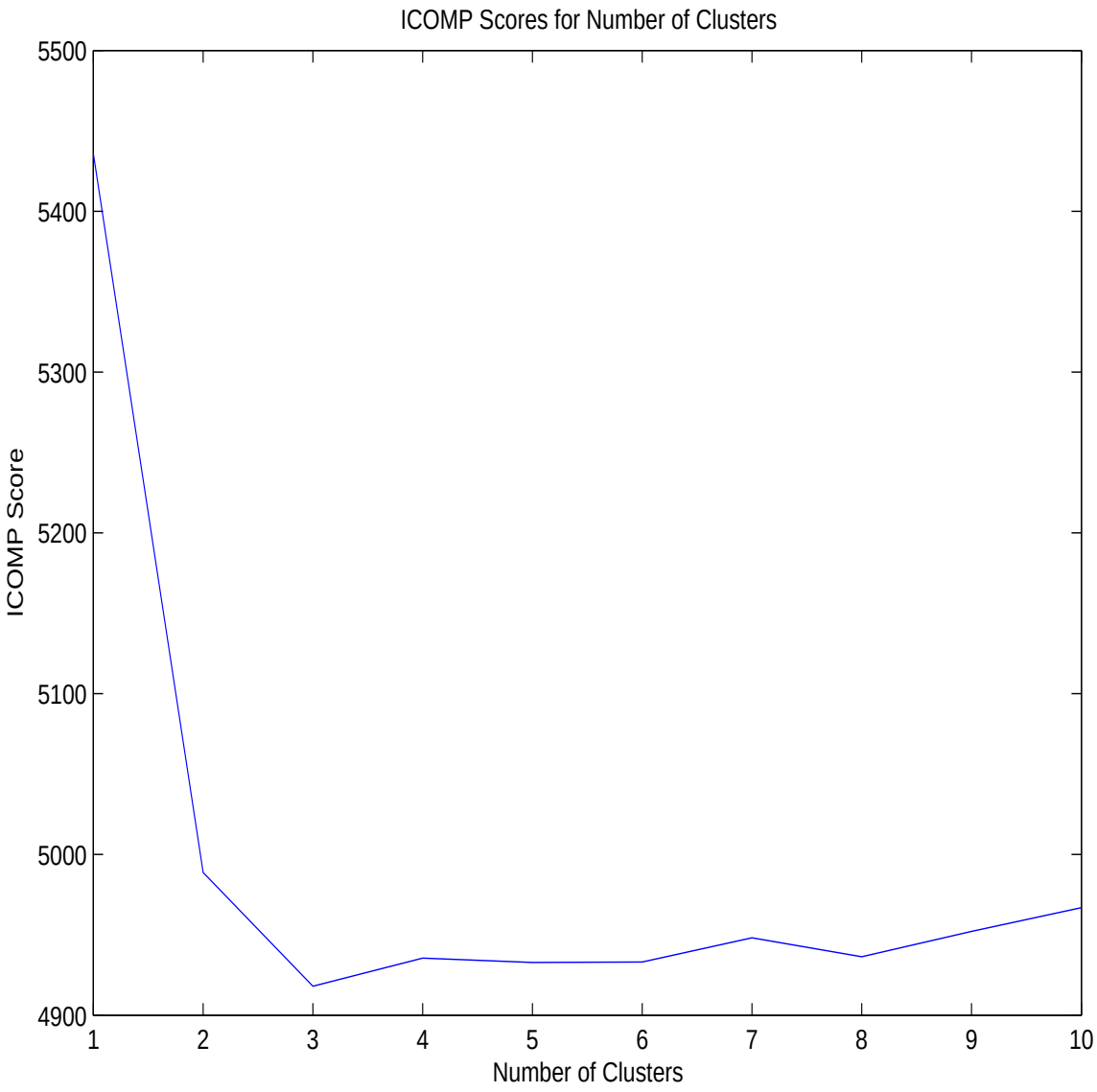

Figure 10.2: Plot of ICOMP Scores for Simulated Data Set 10-1.

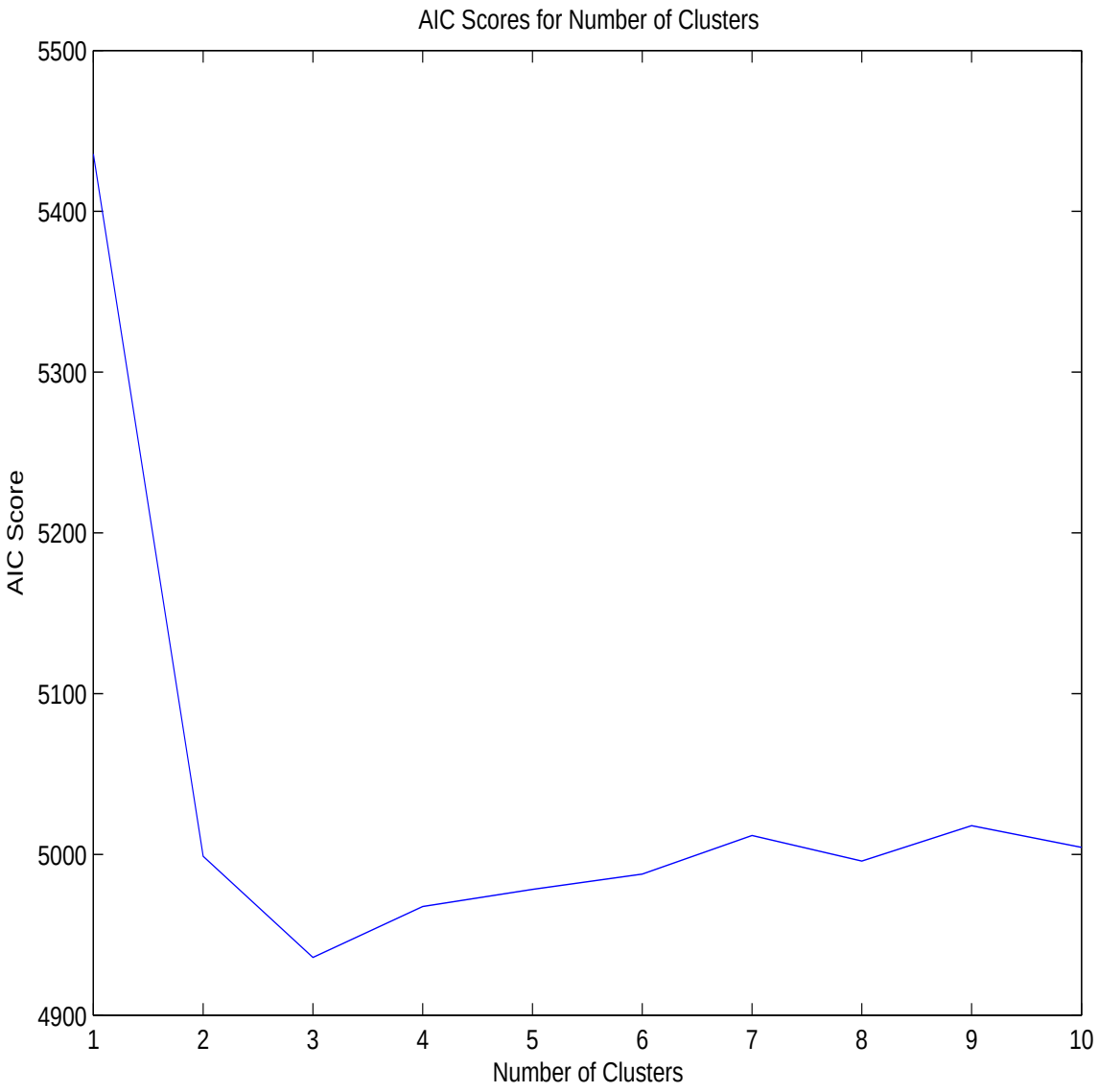

Figure 10.3: Plot of AIC Scores for Simulated Data Set 10-1.

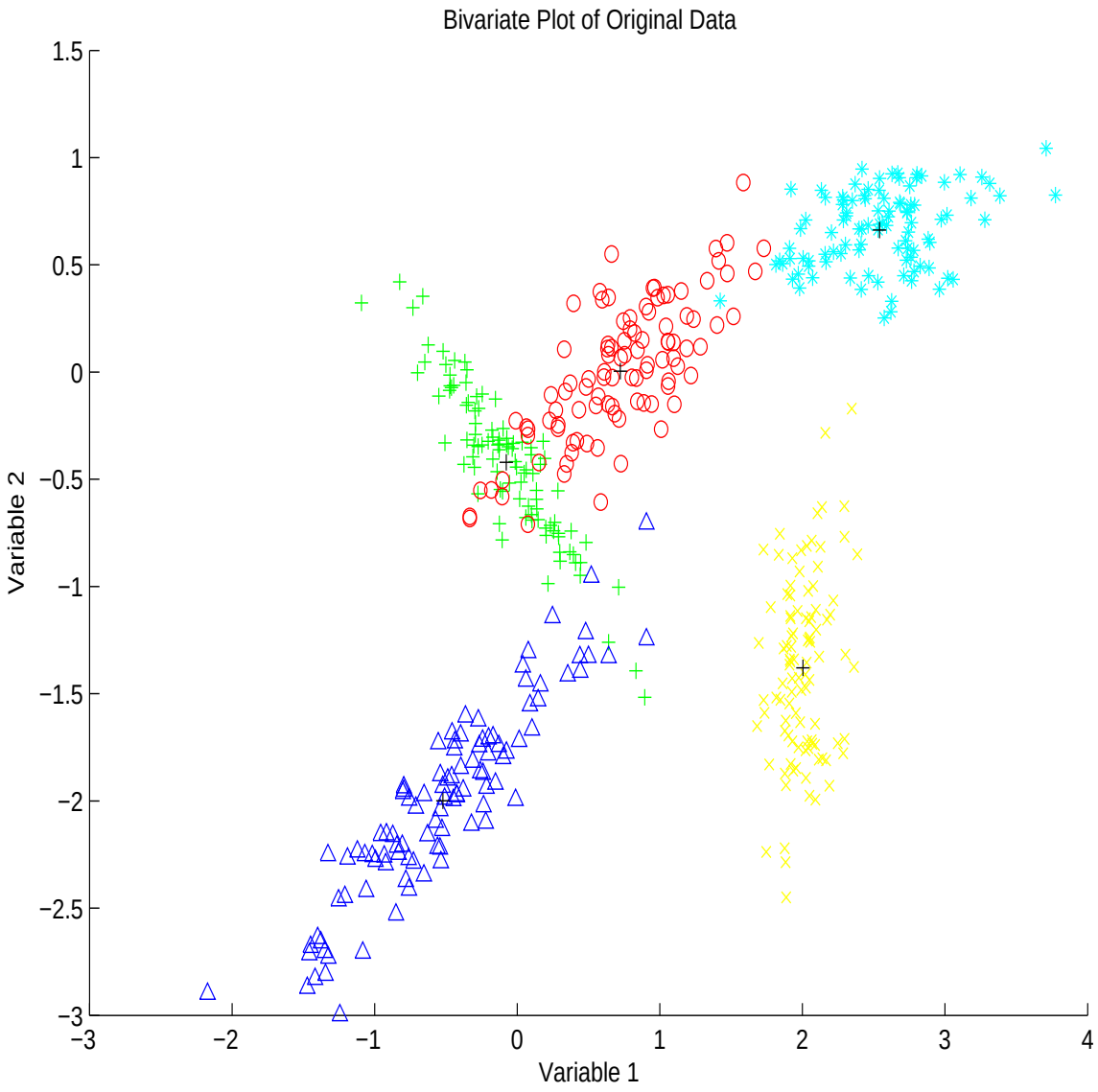

Figure 10.4: True Classification of Simulated Data Set 10-2.

Table 10.3: ICOMP and AIC Scores for the Mixture Models in Simulated Data Set 10-3.

Number of Clusters	ICOMP Score	AIC Score
1	2473.1	2480.5
2	2313.8	2360.1
3	1572.1	1556.1
4	1577.8	1558.7
5	1614.3	1591.8
6	1635.6	1607.2
7	1644.6	1606.0
8	1682.2	1642.4
9	1703.7	1640.8
10	1698.2	1617.6

in the ICOMP expression can use these estimates to give accurate scores. Figures 10.5 and 10.6 show plots of the ICOMP and AIC scores for this data set respectively.

10.5.3 Example 3

This trial used 500 simulated data points in 3 clusters which overlap and are ellipsoidal. The first cluster has 150 data points while the second cluster has 250 data points and the third cluster has 100 data points. Figure 10.7 shows the different clusters.

We used the GEM algorithm with GARM initialization to analyze this data set. A listing of the information scores for the respective number of clusters is given in table 10.3. Figures 10.8 and 10.9 show the plots of the ICOMP and AIC scores respectively.

The results of this trial found that both ICOMP and AIC identified $n = 3$ as the correct number of clusters. Because these clusters overlap and show ellipsoidal covariance structure, we believe that it would be difficult to reach these correct results with a seed-based analysis algorithm.

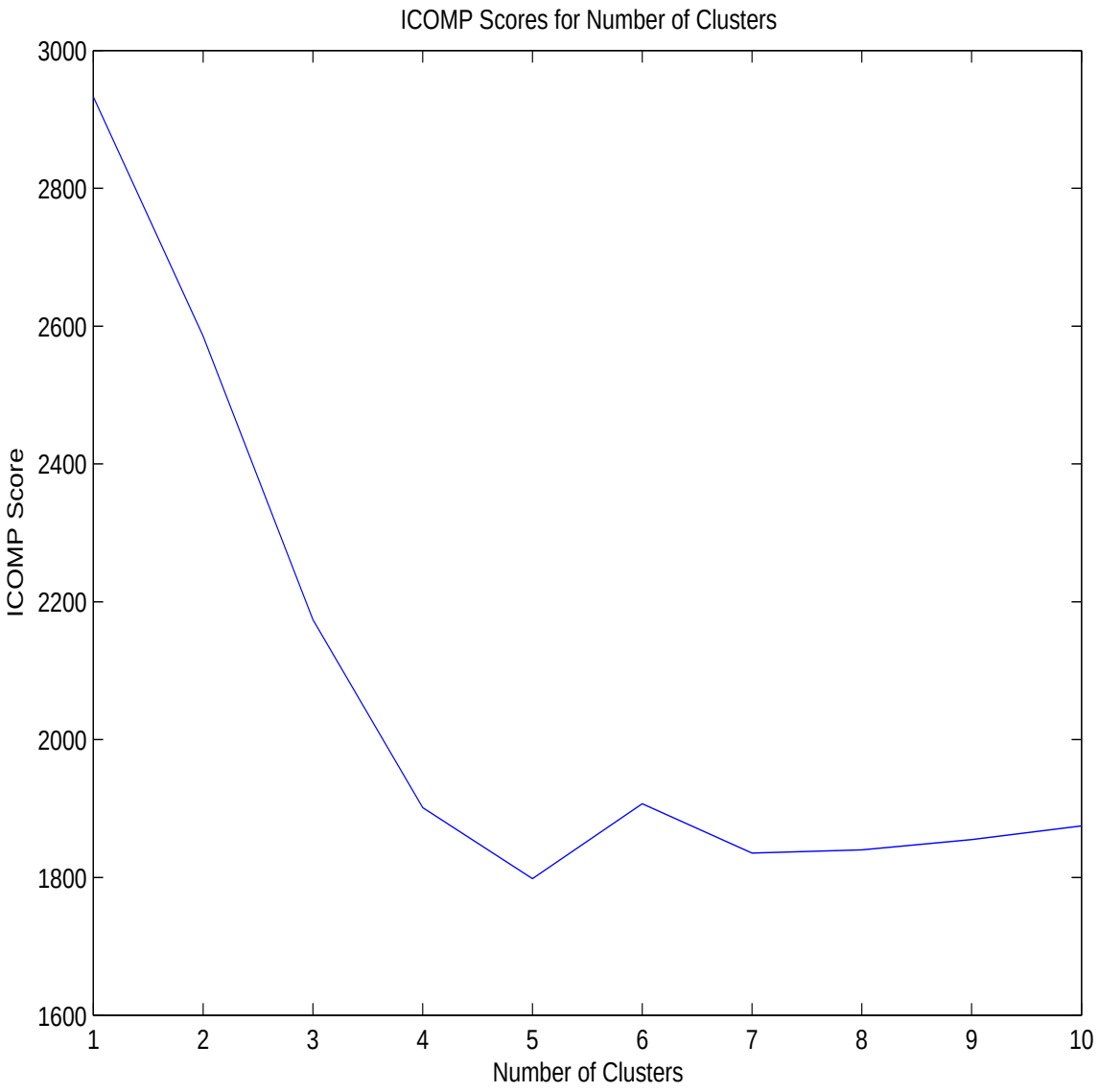

Figure 10.5: Plot of ICOMP Scores for Simulated Data Set 10-2.

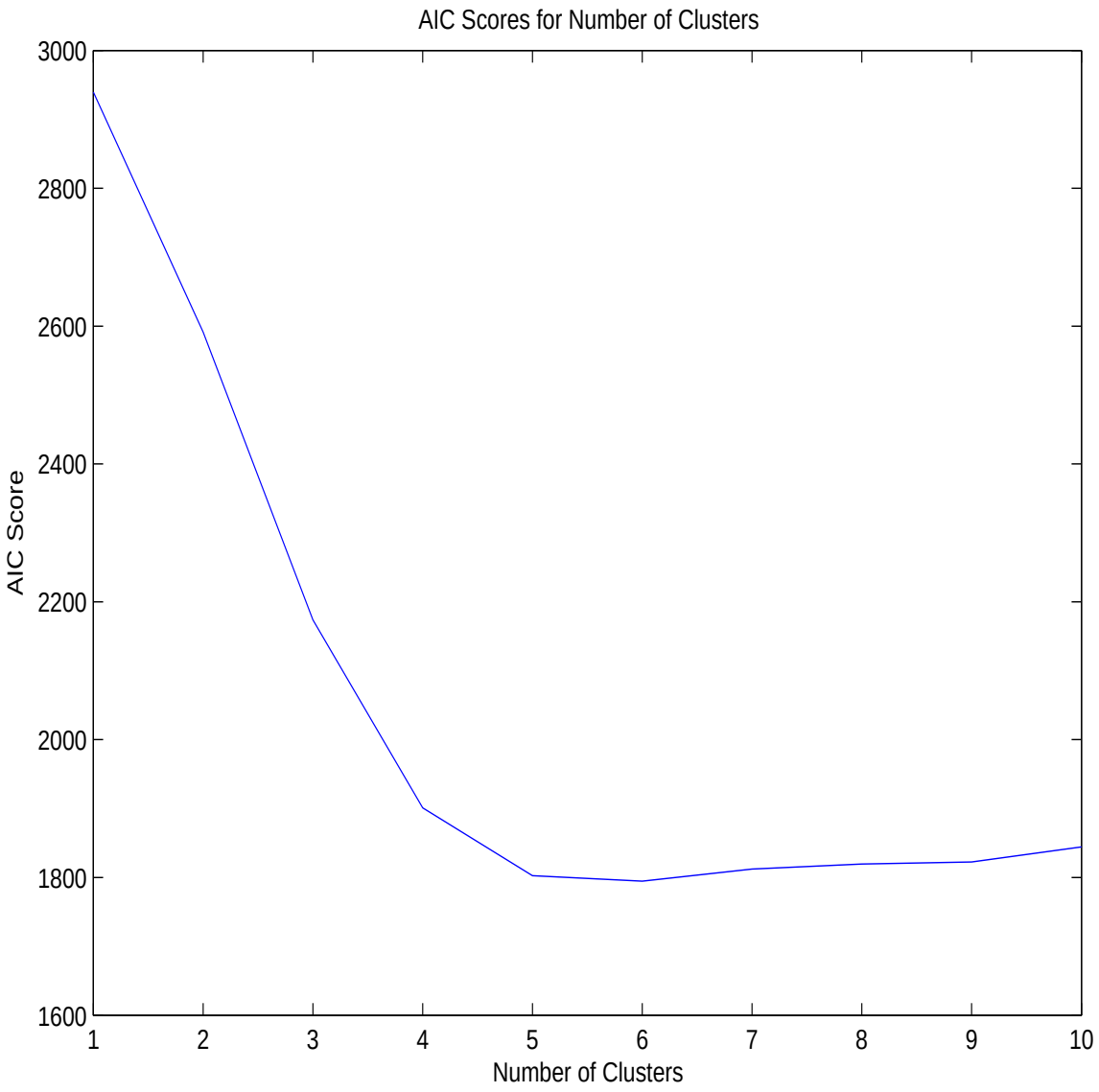

Figure 10.6: Plot of AIC Scores for Simulated Data Set 10-2.

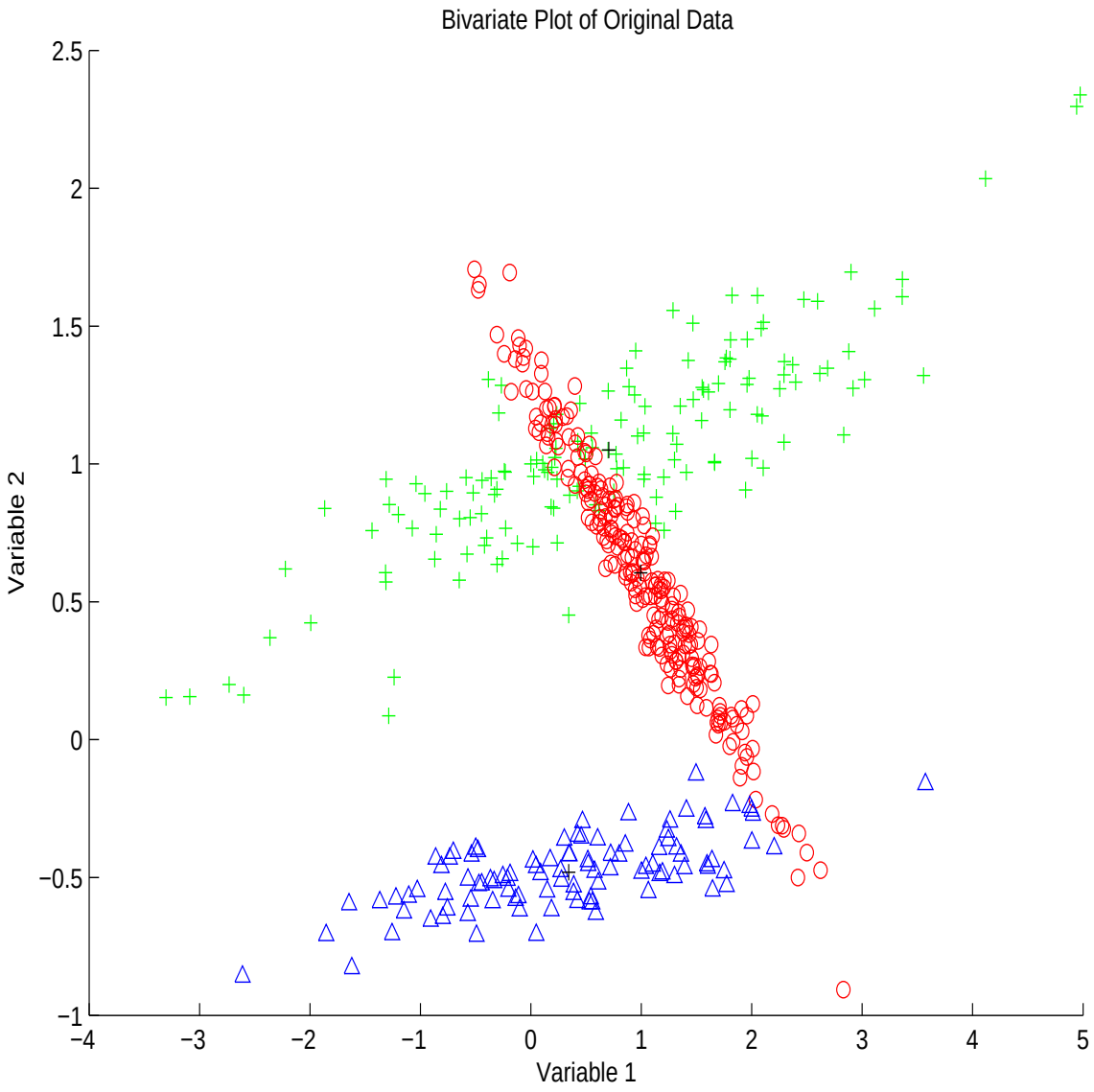

Figure 10.7: True Classification of Simulated Data Set 10-3.

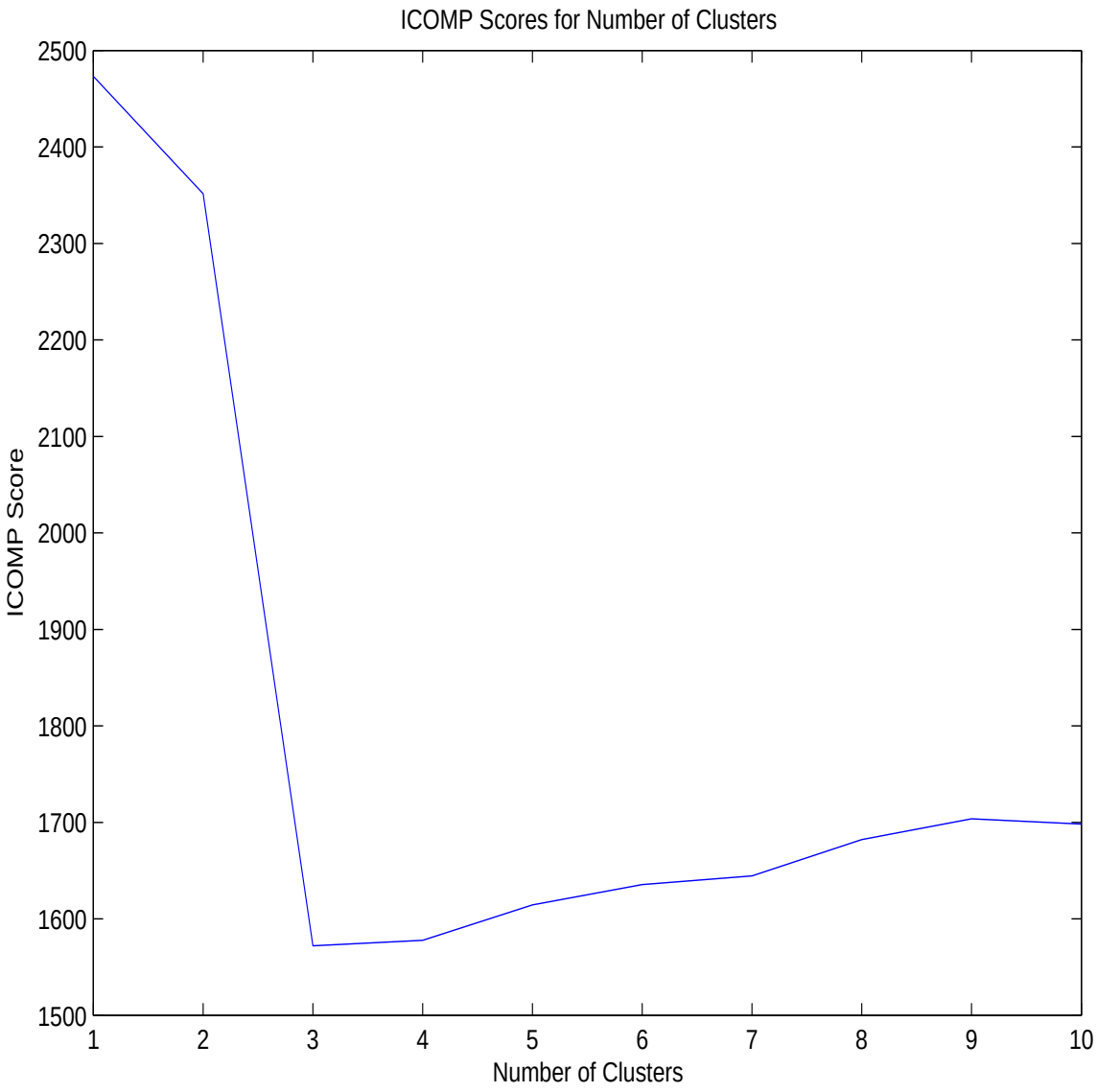

Figure 10.8: Plot of ICOMP Scores for Simulated Data Set 10-3.

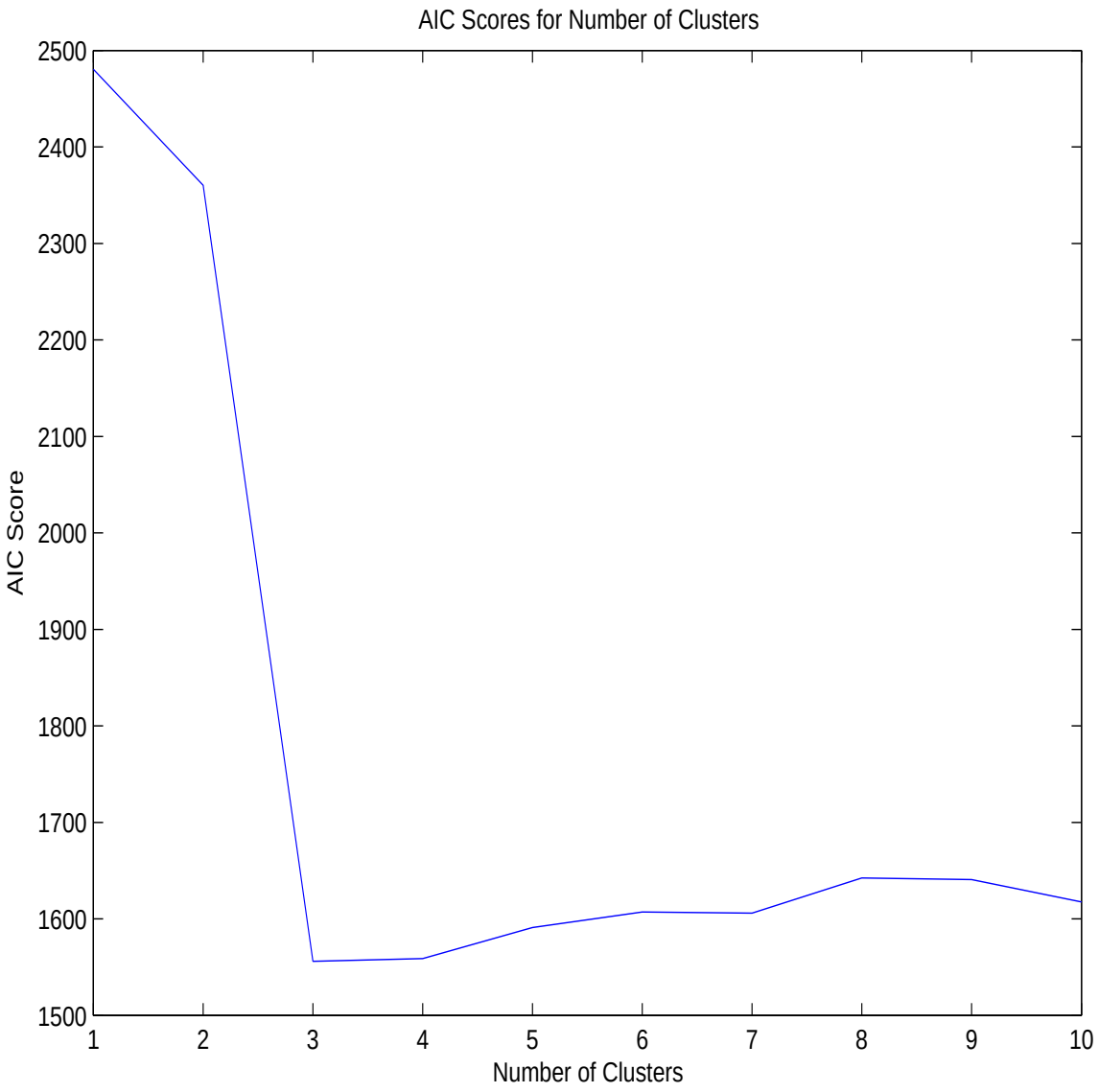

Figure 10.9: Plot of AIC Scores for Simulated Data Set 10-3.

10.6 Conclusion

This chapter showed how the GEM algorithm can be used in an information scoring routine. It also showed how this method can correctly identify the best number of cluster components in a multivariate mixture modeling study. We believe that this new method can be applied to many complex data analysis and pattern recognition studies.

Chapter 11

Mixture Models in Astronomy

11.1 Introduction

In recent years, due to improvements in detector and telescope technology, there has been an explosion of data generated in astronomical research (Szalay and Gray 2001). In order to manage this flood of information, data mining methods have become indispensable tools in astronomical research. Vast amounts of astronomical data are being archived each year. Efficient and accurate data mining methods must be developed and implemented to deal with the onslaught of astronomical data. An important problem emerging from astronomical sky surveys is classification. For example, Zhang and Zhao (2003) describe the problem of classifying stars, galaxies, and active galactic nuclei based on their photometric parameters. Accurate, automated methods of classifying these kinds of objects select candidates for future scrutiny and can identify patterns and relationships among the targets being studied that may not be apparent to human researchers (Zhang and Zhao 2003).

The last decade saw the rise of astronomical survey data. Projects like the Sloan Digital Sky Survey (SDSS), the Two-Micron All Sky Survey (2MASS), the Digitized

Palomar Sky Survey, as well as catalogs from facilities like the U.S. Naval Observatory (USNO), generated terabytes of information automatically recorded by robotic telescopes. With the exponential increase in computing power, during the last two decades, researchers have implemented statistical algorithms in astronomical research. One of the primary data processing methods is Principle Component Analysis (PCA). Mathematically, principle components (PC's) are eigenvectors of the covariance matrix, which form orthogonal linear combinations of the parameters. PCA takes an observed set of data and returns a linear combination of uncorrelated variables (Kendall 1957, Kendall and Stewart 1966). PCA can be used for both data compression and classification (Murtagh and Heck 1987). Most of the variance of a data set is usually contained in relatively few PC's, so many PC's can be deleted with little loss of information. Lawrence (1987) states that one of the main uses of PCA is to find correlations between the input parameters of a data set, which acts to compress the data set and reduce the dimensionality.

During the last decade, methods of classification in astronomy based on artificial intelligence have also been explored. Artificial Neural Networks (ANN's) have been used for star/galaxy classification (Odewahn et al. 1992), galaxy morphology (Storrie-Lombardi et al. 1992) and classification of stellar spectra (Bailer-Jones et at. 1998). More recently, Support Vector Machines (SVM) have been implemented in astronomical research (Wozniac et al. 2001, Humphreys et al. 2001). This type of classification separates classes of objects using hyperplanes in high-dimensional space (Vapnik 1995). Likewise, Learning Vector Quantization (LVQ) is a supervised version of Kohonen's Self-Organizing map, where input vectors are mapped into a set of weightvectors so that topology is preserved. LVQ is especially useful for reducing the dimensionality of data. Bazell and Peng (1998) pioneered the use of LVQ in astronomical data.

Implementing classification algorithms in a multi-stage process can improve their accuracy. Zhang and Zhao used PCA as a preprocessing method for both SVM and LVQ. In their studies, they collected data from sources like USNO and 2MASS to form a 10-variable data set of stars, galaxies, and active galactic nuclei (AGN) with 5547 observations. They first applied PCA to this data set, deleting PC's that contained little information. The preprocessed data then became the input to LVQ and SVM. Using this two-step process, Zhang and Zhao achieved classification accuracies above 89% for both LVQ and SVM. Their numerical trials tried to separate stars from galaxies and AGN's, and also AGN's from normal galaxies (Zhang and Zhao 2003).

In order to complement the artificial intelligence analysis methods, we processed some astronomical data sets with the information scored GEM algorithm. This method not only finds near-optimal cluster assignments for the data sets under the mixture modeling paradigm, but also uses information criteria scoring functions to calculate the best number of mixture components. Information scored analysis provides an objective criteria to judge the best mixture model for a data set, overcoming any possible subjective bias of human researchers. In addition, unlike the artificial intelligence approaches, mixture modeling does not need to be trained. Whereas the artificial intelligence methods focuses on matching data points to prototype examples, the mixture modeling approach analyzes the covariance structure of the data set. This chapter will present the results of the scored mixture model analysis of two astronomical data sets. Section 2 will show the results from the analysis of stellar kinematic data of Soubiran (1993). The stellar kinematic data set examined velocities of stars in our galaxy relative to the galactic center. Using our scored mixture modeling analysis, we will analyze this data set, testing the hypothesis about how many populations of stars inhabit the Milky Way. In Section 3, we apply our information scored GEM algorithm to the Zhang and Zhao (2003) data set. We will analyze the

scored cluster calculations of the different components, looking for the best number of clusters within this data set. The chapter concludes in Section 4.

11.2 Stellar Kinematic Data

The first data set that we processed was the stellar kinematic data set of Soubiran (1993). This data set studied the proper motions of stars in our galaxy. Soubiran collected data about the motion of stars towards the poles of our galaxy (the V component) and the motion of stars about the eventual radial velocities (the U component). These data come from a survey of 7 square degrees near the globular cluster M3. Soubiran studied proper motions of stars from Schmidt photographic plates that spanned 40 years. She then used the small changes in positions of these stars to calculate the components of velocities relative to the galactic center.

Although the historical paradigm of galactic structure has two populations of stars, which are the disk and the halo, since the 1990's, researchers have found evidence of three populations. These three populations are the thin disk, the thick disk, and the halo stars, which differ in in their spatial distributions, metallicities, and kinematics. A bivariate plot of the Soubiran data set (figure 11.1) does not show any strong tendency towards two or three components. Bensmail et al. (1997) applied a Bayes factor analysis method to this data set. The conclusion of that study was that there are three populations of stars in this data set instead of two.

As another test of the possible number of stellar components, we applied our GEM algorithm with information scores. We calculated the information scores for two components and three components. Table 11.1 shows the score results. Figures 11.2 and 11.3 show the classifications of this data set with the respective number of mixture components, where different colors and symbols denote different classifications.

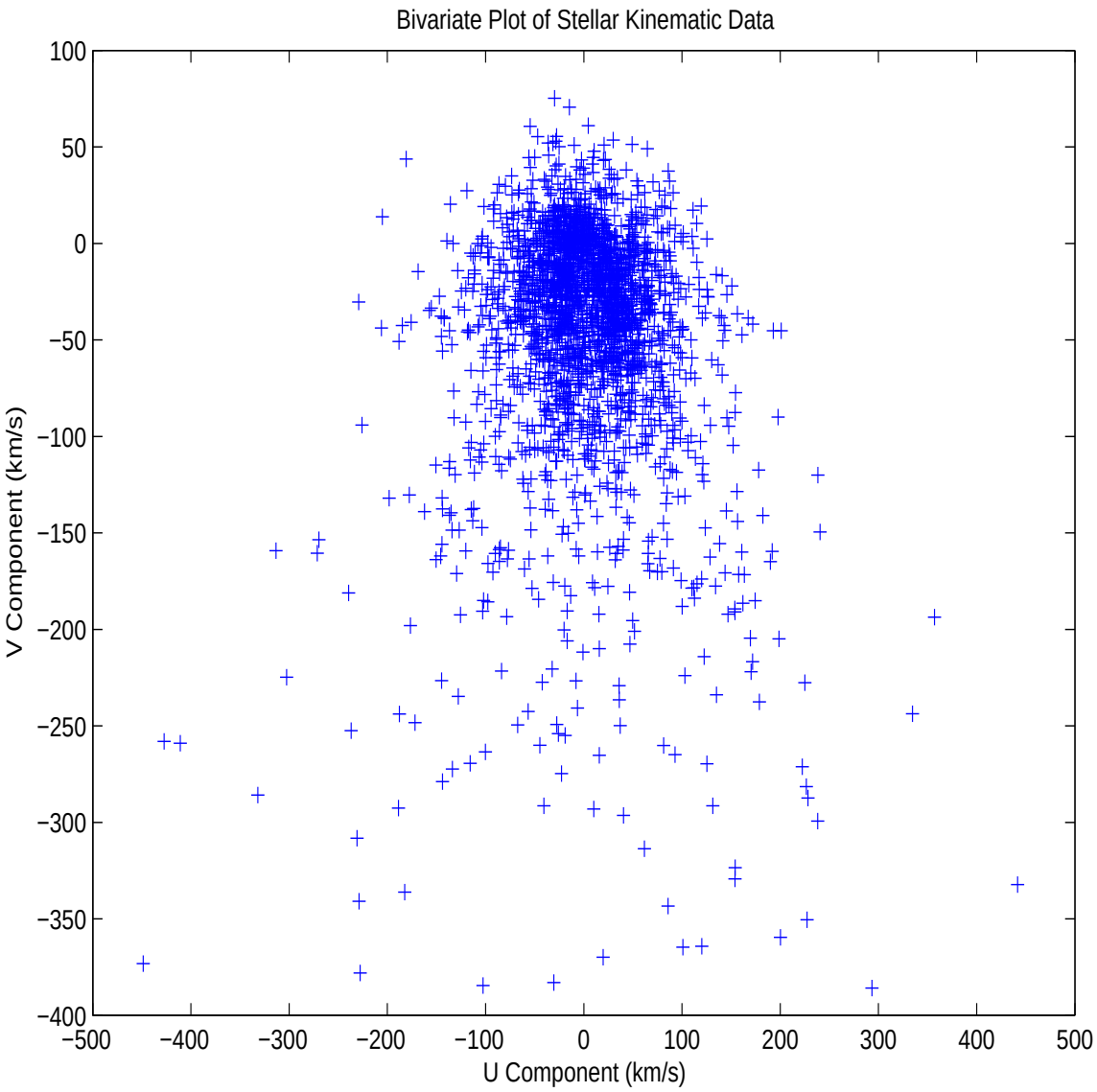

Figure 11.1: Plot of Stellar Kinematic Components.

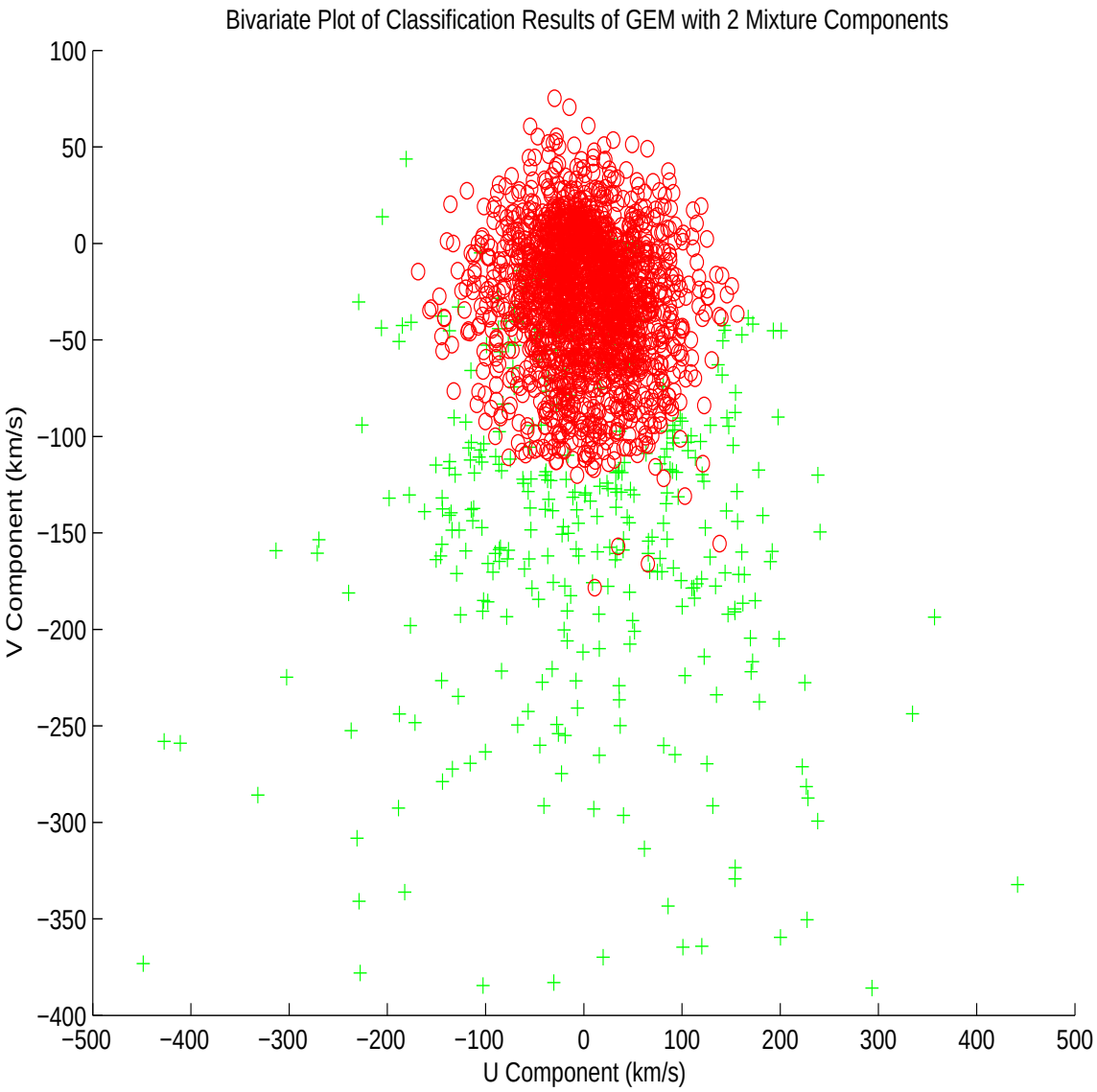

Figure 11.2: Plot of 2 Mixture Components of Stellar Kinematic Data.

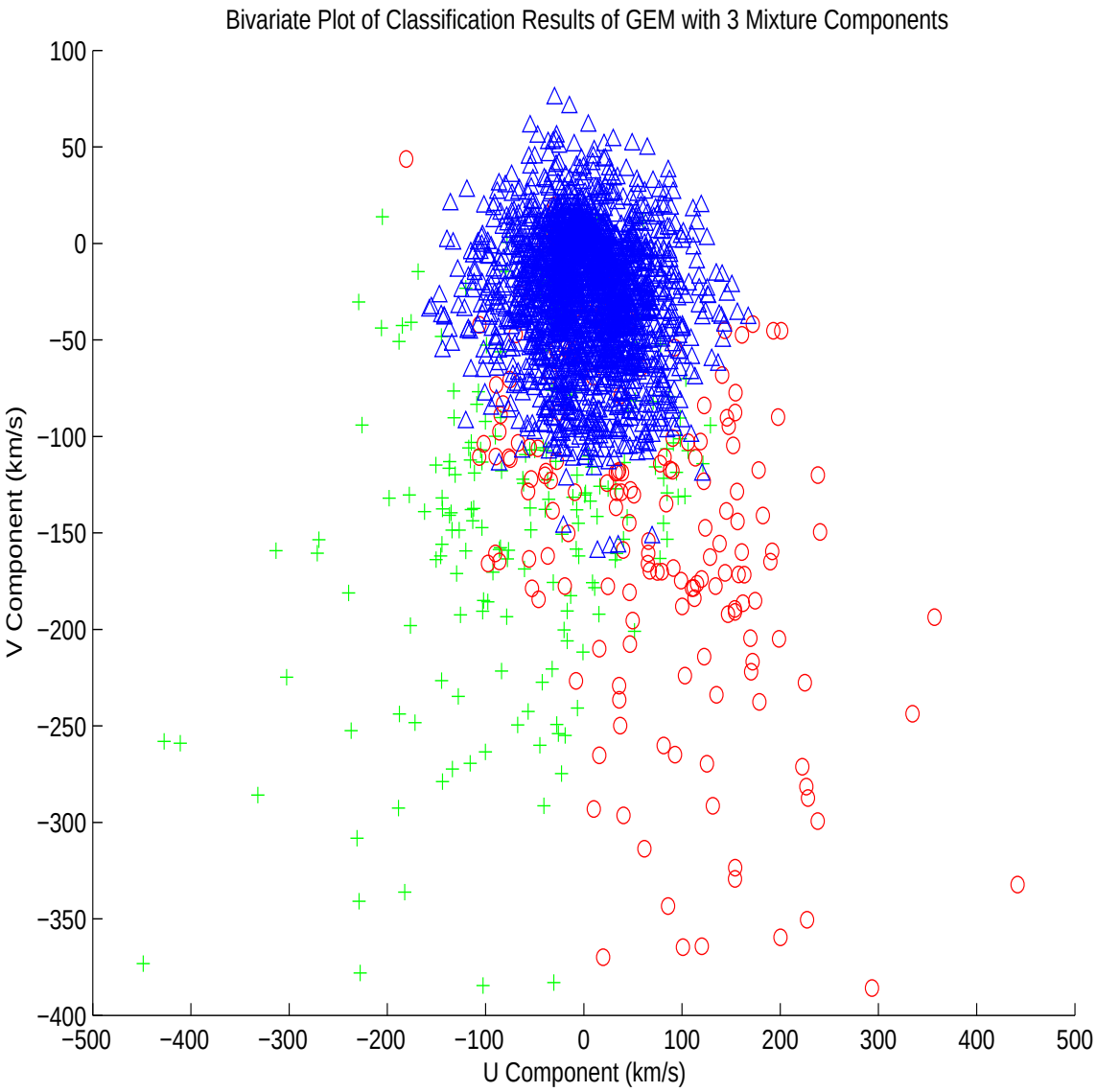

Figure 11.3: Plot of 3 Mixture Components of Stellar Kinematic Data.

Table 11.1: ICOMP and AIC Scores for the Stellar Kinematic Data with Different Numbers of Clusters.

Number of Clusters	ICOMP Score	AIC Score
2	51044.2	51024.4
3	51025.5	51007.6

We can see that the minimum information scores for both ICOMP and AIC occurs at 3 clusters. This result gives further evidence that there are in fact 3 populations of stars in our galaxy instead of 2 populations.

11.3 Astronomical Survey Data

The next data set that we considered was from Zhang and Zhao (2003). These authors explored how automated classification methods can be used to classify astronomical data. They are especially interested in classification methods in astronomy because of the upcoming survey telescope LAMOST. The Large Area Multi-Object Spectroscopic Telescope (LAMOST) is currently being constructed in China and is scheduled to begin operation in 2007. LAMOST will be one of the largest survey telescopes ever built and it is expected to produce over ten-million spectra of a wide variety of objects, including quasars, galaxies and stars. The data archive, expected to exceed 1 terabyte in size, will join other data already housed at the National Astronomical Observatory of China (NAOC) as part of the Virtual Observatory of China (Luo et al. 2004). Zhang kindly gave us a copy of this data while I was visiting NAOC under the 2004 NSF Summer Institute in China exchange program.

In order to deal with the volume of data that will be generated by LAMOST, accurate and efficient algorithms need to be implemented. The survey studies conducted by LAMOST will rely on classification methods developed in statistical and computational research. LAMOST is one of the major scientific projects undertaken

by the Chinese Academy of Science. It will be a spectroscopic telescope with a 5 degree field of view that will be able to record up to 4000 spectra simultaneously. LAMOST will be a quasi-meridian transit telescope employing a 4 meter mirror that can record spectra as faint as magnitude 20. It is estimated that LAMOST will record one-dimensional spectra for 10^6 stars, 10^7 galaxies, and 10^6 QSO's, with a spectral range of 3700 to 9000 Angstroms. Because of its unique design and large field of view, LAMOST will have one of the fastest data acquisition rates of any spectral telescope in the world. Although LAMOST sits atop a remote mountain at the Xinglong observing station, the generated data will be stored and processed at NAOC in Beijing, where it will be made available to the scientific public via a user-friendly web interface. LAMOST will be a world-class observing facility that will make major contributions to wide-field astronomy (Luo et al. 2004).

The Zhang and Zhao data set consisted of 5547 datapoints with 10 dimensions that are parameters describing the objects in the X-ray, optical, and infrared bands. The X-Ray sources were compiled from the ROSAT All-Sky Survey (RASS) Bright Source Catalog, and the RASS Faint Source Catalog. The optical band observations came from the USNO-A2.0 catalog, while the infrared band was taken from the 2MASS database. The parameters used in this study denote the intensities of the sources in the different wavelengths. These authors also show that PCA preprocessing is an effective way to reduce the dimensionality of this data set by calculating numerical trials with 3, 4, 5, and 6 PC's. In their trials that try to separate stars from AGN's and galaxies, they show that using from 3 to 6 PC's of the data still allows classification accuracies of 94.9% and above.

We applied the information scored GEM algorithm to both the entire data set and to the stars, galaxies, and AGN subsets separately. Like the original authors, we used PCA preprocessing. In our trials we used only the first 3 principle components

of the data set. These three principal components contain 86.35% of the variance of the data, so there is little loss of information with the reduced dimensionality. We ran scoring trials with increasing numbers of cluster components in these data sets until we were satisfied that we found the minimum information scores for each case. We used the scoring criteria with Maximum Likelihood/Emperical Bayes covariance estimator regularization to prevent possible singularities. This let us assign information scores to the different mixture model cases, with the only condition on the cluster component being that each cluster must have at least one assignment. The results of the respective scoring trials follow.

11.3.1 Galaxy Data Subset

The galaxy data subset had only 173 data points. The results of the scored the GEM mixture model trials are given in table 11.2. Plots of these scores are given in figures 11.4 and 11.5. We can see that ICOMP showed a minimum score at 3 clusters, whereas AIC showed a minimum at 12 clusters. We can see from the plots that ICOMP shows a distinct minimum score at 3 clusters. We believe that this strong difference between the different scoring functions indicates very complex covariance structure in this data subset. Because ICOMP is better able to capture the complex interactions between covariance parameters, we believe that the ICOMP score of 3 indicates the best number of mixture model components in this data subset.

11.3.2 Active Galactic Nuclei (AGN) Data Subset

The AGN data subset has 1656 data points. Zhang and Zhao (2003) indicate that this sample contains 909 quasars, 135 BL Lac objects, and 612 active galaxies. Our calculations show that the minimum ICOMP score occurs at 13 cluster components, and the minimum AIC score is at 19. We again believe that this data set has a

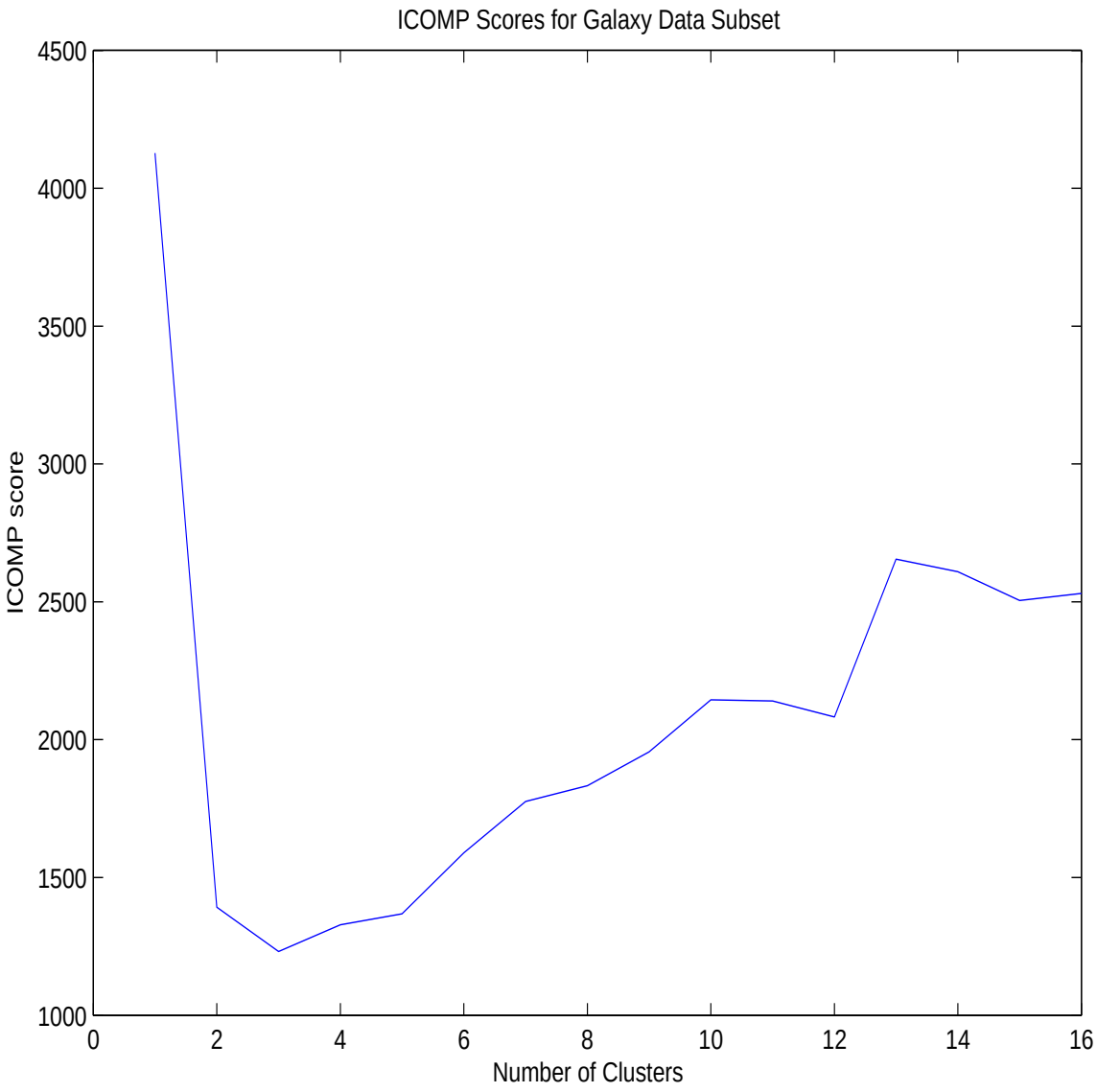

Figure 11.4: Plot of ICOMP Scores for Galaxy Data Subset.

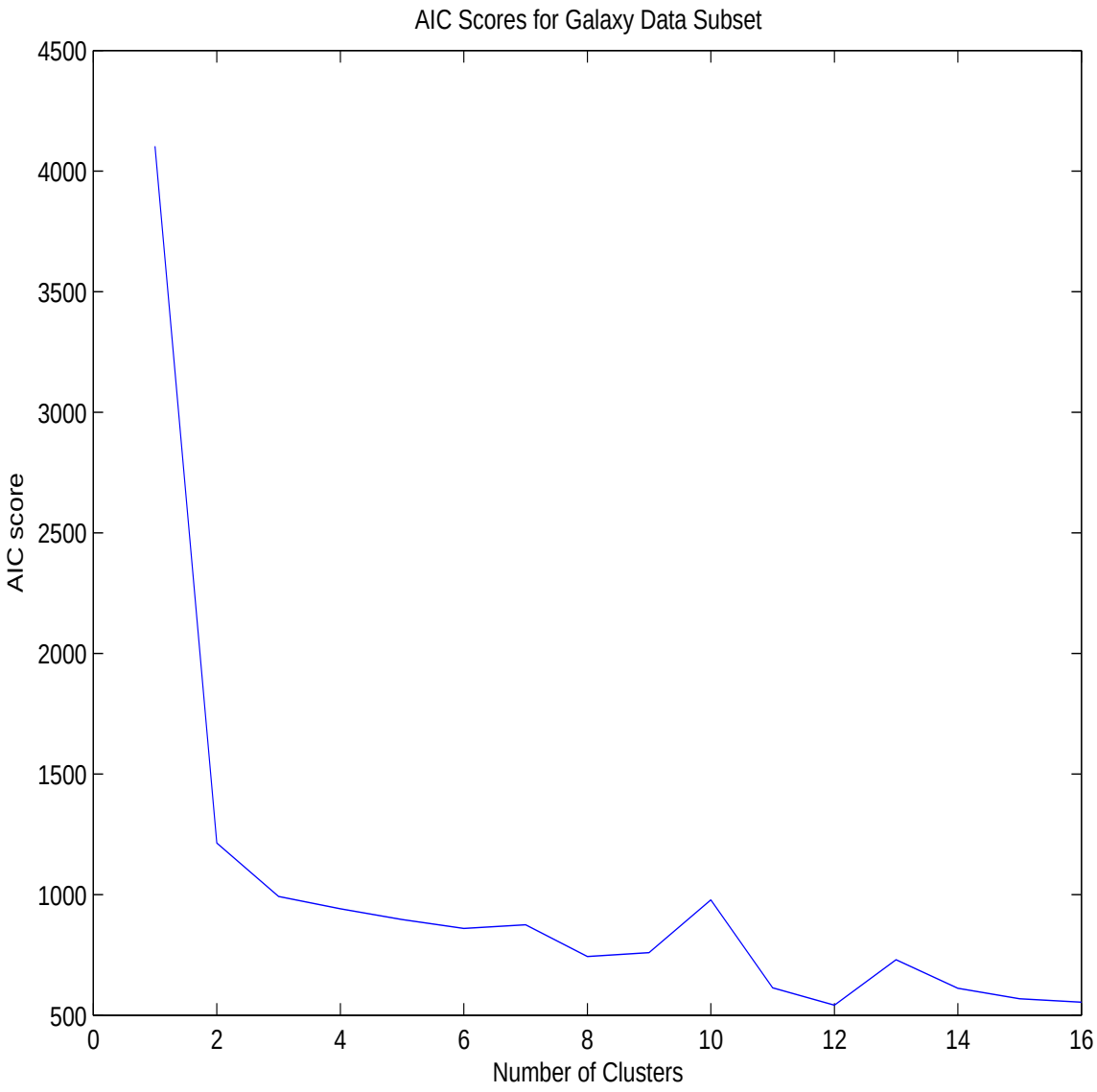

Figure 11.5: Plot of AIC Scores for Galaxy Data Subset.

Table 11.2: ICOMP and AIC Scores for the Galaxy Data Subset with Different Numbers of Clusters.

Number of Clusters	ICOMP Score	AIC Score
1	4127.9	4102.4
2	1392.1	1214.5
3	1231.8	992.4
4	1328.1	942.0
5	1368.3	897.3
6	1589.6	860.1
7	1776.1	875.1
8	1832.5	743.3
9	1955.4	759.6
10	2144.1	978.2
11	2140.5	613.9
12	2082.0	542.3
13	2654.7	730.8
14	2609.0	611.9
15	2505.0	568.6
16	2530.5	554.4

complex covariance structure, but that the minimum score indicated by ICOMP of 13 is the correct number of mixture components. Our information scoring trial results are given in table 11.3. Figures 11.6 and 11.7 show the ICOMP and AIC scores for this data set respectively.

11.3.3 Star Data Subset

This data set had 3718 star observations, which included normal stars, cataclysmic variables, and white dwarfs. Table 11.4 shows the information scores for the respective number of mixture components in this data set. We can see that the minimum ICOMP score occurs at 15 clusters, while the minimum AIC score is at 24 clusters. Figures 11.8 and 11.9 show plots of these scores for the cluster trials. On close examination, the ICOMP scores show a slight upward trend after the minimum, whereas AIC does not really show any trend. We believe that ICOMP returns the best mixture

Table 11.3: ICOMP and AIC Scores for the AGN Data Subset with Different Numbers of Clusters.

Number of Clusters	ICOMP Score	AIC Score
1	29936.4	29931.5
2	11632.4	11523.0
3	9029.7	8874.9
4	8437.2	8224.5
5	8412.9	8154.7
6	8585.1	8087.1
7	9009.7	8601.1
8	6215.9	5643.1
9	8511.3	7944.3
10	6262.3	5513.0
11	6459.8	5467.4
12	6384.9	5519.4
13	5925.0	4628.0
14	6084.4	4950.7
15	6687.6	5501.6
16	6866.8	5397.1
17	7536.7	5912.0
18	6272.2	4645.8
19	6467.0	4435.5
20	7065.6	5488.7

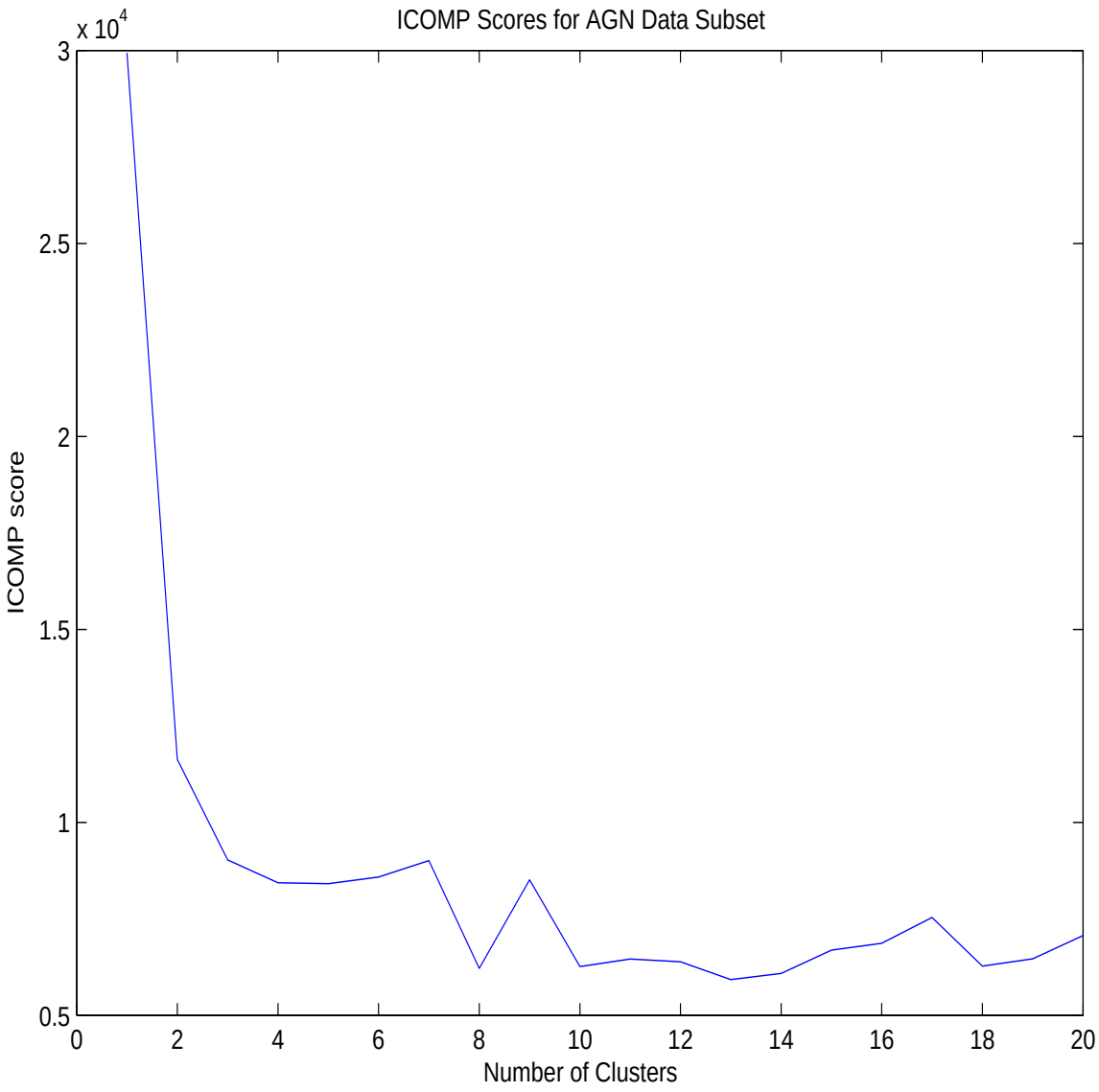

Figure 11.6: Plot of ICOMP scores for AGN data subset.

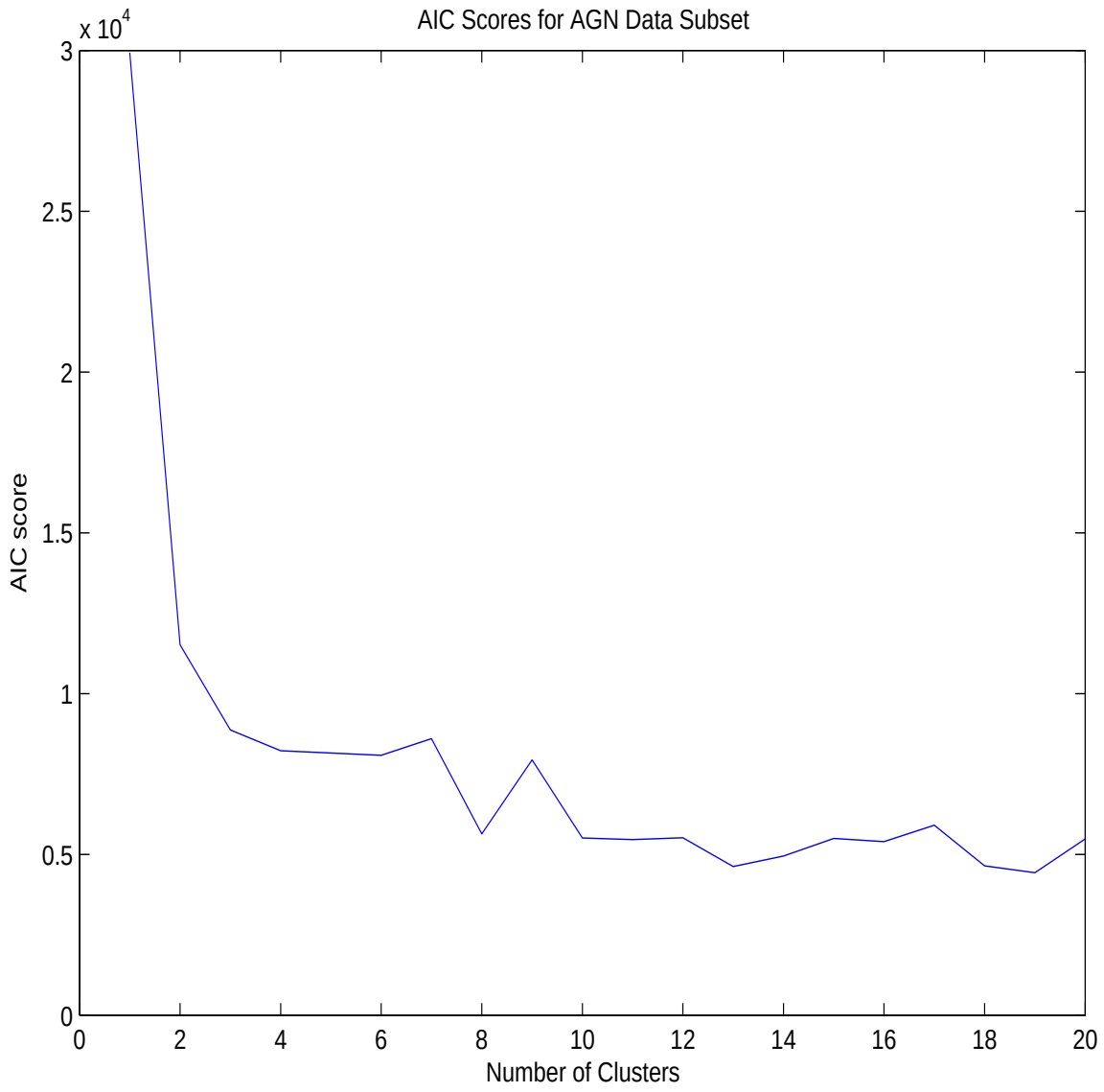

Figure 11.7: Plot of AIC Scores for AGN Data Subset.

Table 11.4: ICOMP and AIC Scores for the Star Data Subset with Different Numbers of Clusters.

Number of Clusters	ICOMP Score	AIC Score
1	98919.5	98912.3
2	56971.1	56881.0
3	55157.1	55014.6
4	41638.7	41288.8
5	25611.0	25091.6
6	23619.0	23109.7
7	23666.3	22885.5
8	22471.1	21914.3
9	20980.6	20291.5
10	21993.0	20968.0
11	22301.9	20882.5
12	22292.9	21100.5
13	22171.9	20756.5
14	19862.4	18300.1
15	19756.0	18259.6
16	22592.4	20624.0
17	21165.0	19133.5
18	23048.1	20549.7
19	23275.2	20608.7
20	22851.4	20489.4
21	21282.0	18695.5
22	23720.5	20757.9
23	23390.7	20430.0
24	21080.0	17800.2
25	23923.5	20596.1
26	24564.1	20950.3
27	23623.1	19914.5
28	22773.5	19061.2
29	22848.4	19588.8
30	25079.6	20518.3

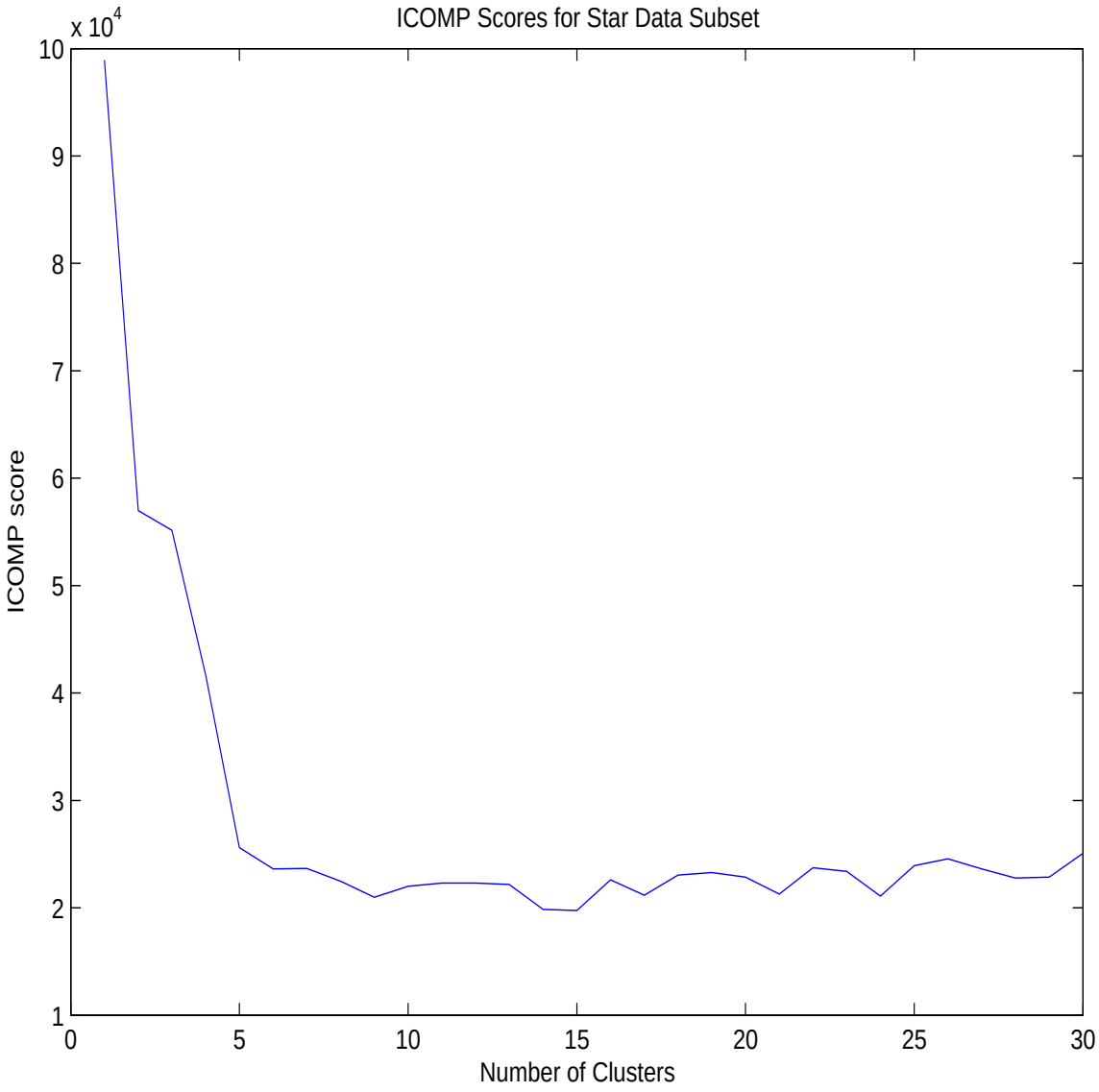

Figure 11.8: Plot of ICOMP Scores for Star Data Subset.

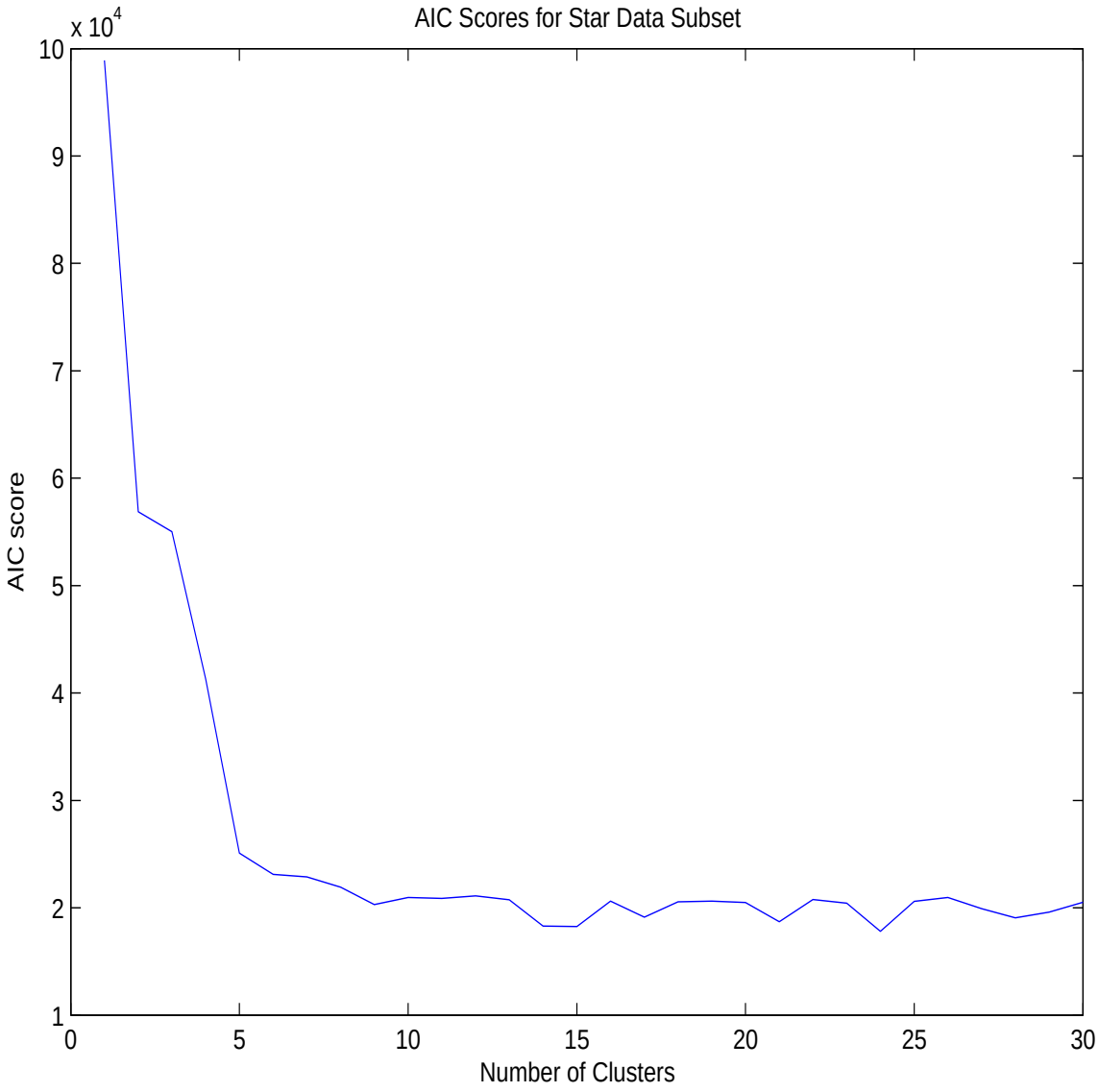

Figure 11.9: Plot of AIC Scores for Star Data Subset.

component calculation, so we regard 15 components as the best number of clusters for this data set.

11.3.4 Complete Dataset

We applied our scored mixture model analysis to the entire data set that included stars, galaxies, and AGN's. Table 11.5 lists the respective information scores for this data set, and figures 11.10 and 11.11 show plots of ICOMP and AIC for this data set respectively. We expected that the optimum number of clusters might be 31 because this is the sum of all of the individual components, but we found that according to the ICOMP scores, the optimal number of components was 17. Interestingly, AIC returned a minumum score of 29, which is much closer to the sum of the components from the other cases. We believe that, because this data set is so complex, ICOMP identified 17 components by merging some of the previously identified components. Considered together, there is no mathematical reason to expect that the number of components will add because the log-likelihoods of the merged data set is distinctly different from those of the data subsets. We believe that this kind of analysis can provide a good starting point for future studies that may try to characterize such complex data.

11.4 Conclusion

In this chapter, we showed how the information scored GEM algorithm can be applied to complex, multivariate astronomical data. The clusters in these data sets overlap and show highly complex structure. We believe that this analysis is the first step in a more detailed study of mixture model cluster analysis in astronomy. Future research will focus on better characterizing the properties of objects identified in the different

Table 11.5: ICOMP and AIC Scores for the Complete Data Set with Different Numbers of Clusters.

Number of Clusters	ICOMP Score	AIC Score
1	145080.0	145075.7
2	81019.0	80933.3
3	82399.7	82270.1
4	75883.9	75584.9
5	75070.3	74737.9
6	56753.8	56169.7
7	54976.0	54388.8
8	52642.6	51987.1
9	47222.3	46443.3
10	48718.1	47820.1
11	49315.0	48176.0
12	42772.0	41720.9
13	45326.3	44324.0
14	43091.2	41342.9
15	53483.3	51436.2
16	43707.3	42288.1
17	42017.9	40572.5
18	48295.1	46065.6
19	46510.1	44051.6
20	47853.0	45063.8
21	45898.5	43546.6
22	47390.4	44630.7
23	46961.5	44084.6
24	42715.3	39661.5
25	48049.6	44459.7
26	45052.4	41555.7
27	46432.9	42957.5
28	45974.0	42314.4
29	42664.0	38840.1
30	46384.0	42182.8
31	47514.1	43768.2
32	44212.0	40109.7
33	44292.8	40045.5
34	44651.6	39674.6
35	48952.7	44400.4
36	49614.4	44222.5
37	45060.4	39656.8
38	45372.5	39998.5
39	49296.3	43450.8
40	46440.4	40587.5

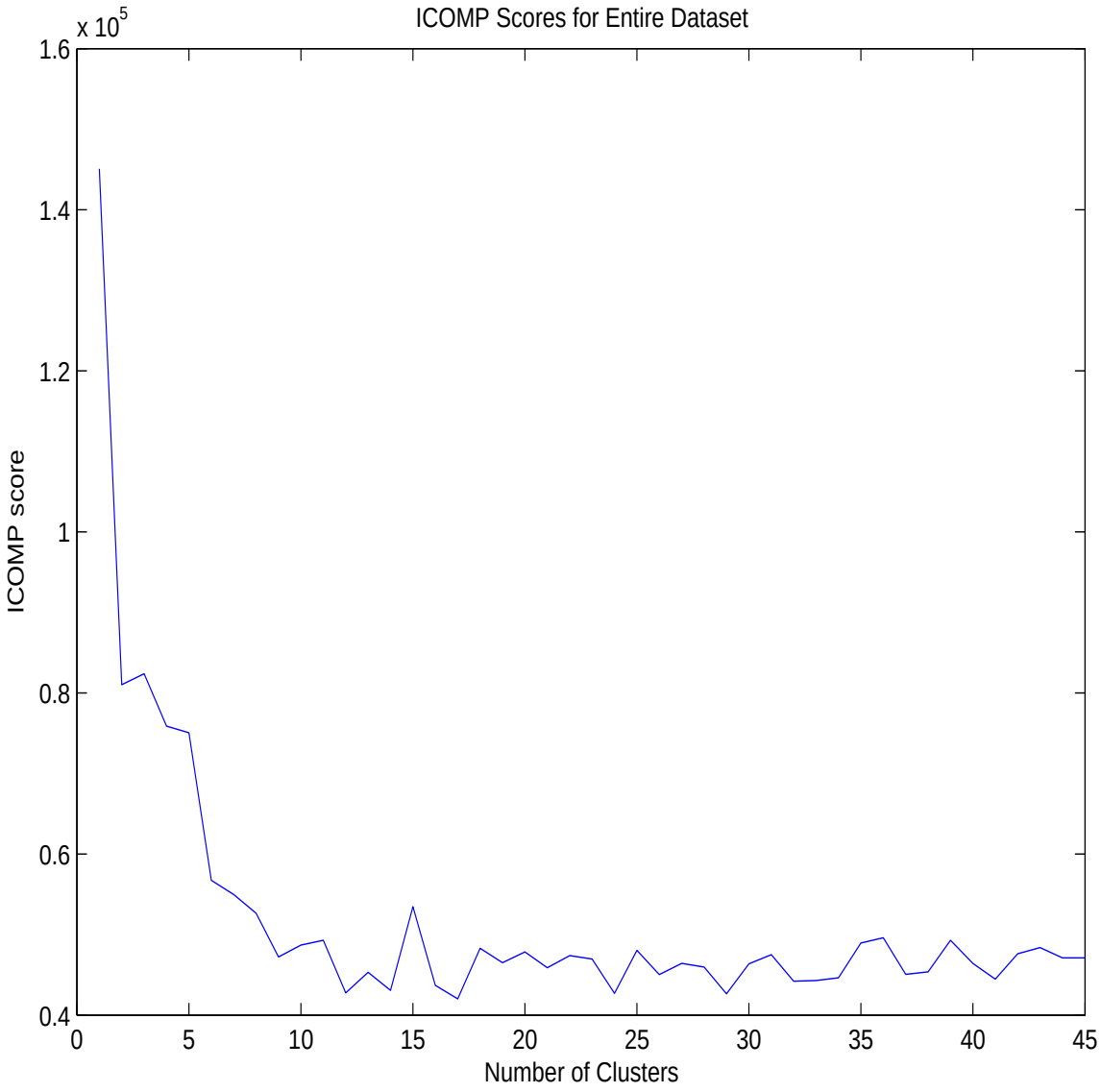

Figure 11.10: Plot of ICOMP Scores for Complete Data Set.

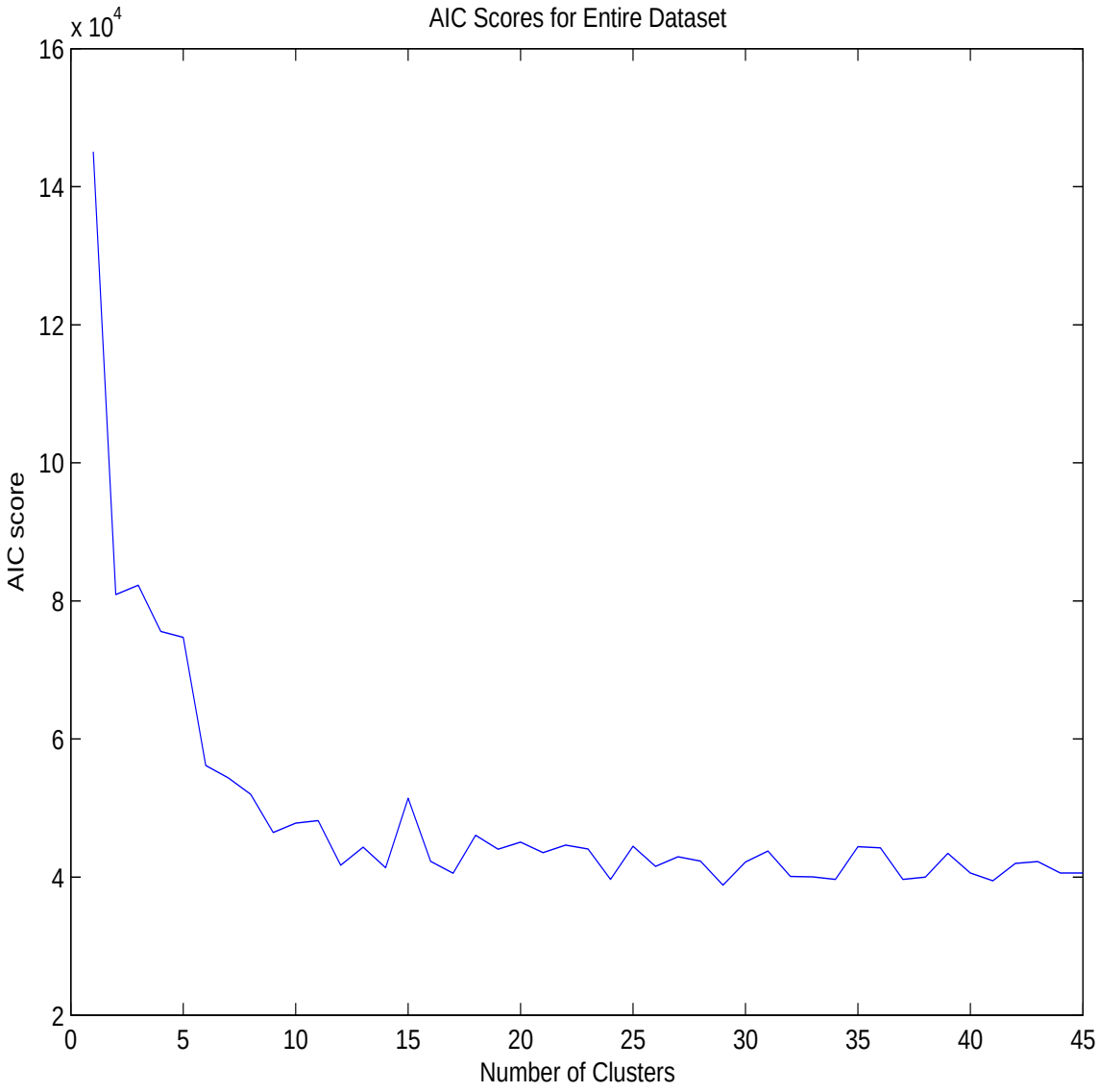

Figure 11.11: Plot of AIC Scores for Complete Data Set.

clusters. Another avenue of future research would be to identify clusters that have few members. These would be unusual objects in parameter space, which would signal objects that need closer scrutiny.

Chapter 12

Conclusion

This dissertation showed how modern analysis methods can be used to process physical data. The first section reviewed classical and modern regression methods, focusing on how modern methods resolve problems inherent to the classical methods. It also reviewed developments in genetic algorithms and the theory behind molecular vibration-rotation spectroscopy. It then showcased examples of using modern regression methods in the analysis of molecular spectra. We believe that modern regression methods can find general applicability in physical science studies that use multivariate linear regression.

The second section explored ways that genetic algorithms can be used in cluster analysis. The results of traditional seed-based methods depend strongly on starting values, whereas genetic algorithm based methods overcome this dilemma. We generalized the Genetic K-means algorithm into GARM, which can accurately identify hyperellipsoidal clusters. In addition, we used the new genetic algorithm based cluster approach in an Expectation-Maximization routine that accurately calculated parameters in multivariate mixture models. We showed how these accurate values can be used in an information scoring method that identifies the best number of components

in a mixture modeling situation. Later, we applied these new analysis algorithms to process multivariate astronomical data. We believe that the genetic algorithm based cluster analysis methodology can be used in many log-likelihood maximization methods and can be implemented in complex data mining and pattern recognition problems.

We hope that other researchers can apply the methods developed in this dissertation to many useful and interesting studies.

Bibliography

AKAIKE, H. (1973) Information theory and an extension of the maximum likelihood principle. IN PETROV, B. N. CSAKI, F. (Eds.) Second International symposium on information theory. Budapest, Academiai Kiado.

AKAIKE, H. (1974) A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control, AC-19, 716-723.

AKAIKE, H. (1978) Time Series Analysis and Control Through Parametric Models. IN FINDLEY, D. F. (Ed.) Applied Time Series Analysis. New York, Academic Press.

AKAIKE, H. (1981) Modern Development of Statistical Methods. IN EYKOFF, P. (Ed.) Trends and Progress in System Identification. New York, Pergamon Press.

AKAIKE, H. (1985) Prediction and Entropy. IN ATKINSON, A. C. FIENBERG, S. E. (Eds.) A Celebration of Statistics: The ISI Centenery Volume. New York, Springer-Verlag.

AMAT, G. NIELSEN, H. H. (1958) Journal of Chemical Physics, 29, 665.

AMAT, G. NIELSEN, H. H. (1961) Journal of Chemical Physics, 34, 339.

AMAT, G. NIELSEN, H. H. (1962) Journal of Chemical Physics, 36, 1859.

AMAT, G., NIELSEN, H. H. TARRAGO, G. (1971) Rotation-Vibration of Polyatomic Molecules, New York, Marcel Dekker.

BABU, G. J. DJORGOVSKI, S. G. (2004) Some Statistical and Computational Challenges, and Opportunities in Astronomy. Statistical Science, 19, 322-332.

BAILER-JONES, C. A. L., IRWIN, M. VON HIPPLE, T. (1998) Semi-automated extraction of digital objective prism spectra. Monthly Notices of the Royal Astronomical Society, 298, 361.

BAZELL, D. PENG, Y. (1998) A Comparison of Neural Network Algorithms and Preprocessing Methods for Star-Galaxy Discrimination. Astrophysical Journal Supplement, 116, 393.

BEALE, E. M. (1969) Cluster Analysis, London, Scientific Control Systems.

BEARSE, P. M. BOZDOGAN, H. (2002) Multivariate regressions, Genetic Algorithms, and Information Complexity: A Three Way Hybrid. IN NISHISATO, S., BABA, Y., BOZDOGAN, H. KANEFUJI, K. (Eds.) Measurement and Multivariate Analysis. Tokyo, Springer.

BEATON, A. E. TUKEY, J. W. (1974) The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. Technometrics, 16, 147-185.

BELLMAN, R. (1961) Adaptive Control Processes: A Guided Tour, Princeton, NJ, Princeton University Press.

BENSMAIL, H. BOZDOGAN, H. (2005) Bayesian unsupervised clustering for mixed data with missing observations. Submitted to JRSS-C.

BENSMAIL, H., CELEUX, G., RAFTERY, A. E. ROBERT, C. (1997) Inference in model-based cluster analysis. Computing and Statistics, 7, 1-10.

BENSMAIL, H., GOLEK, J., SEMMENS, J. HAOUDI, A. (2005) Bayesian Fast-Fourier Transform Based Clustering Method for Proteomics Data. Bioinformatics, 21, 2210-2224.

BHUYAN, L. N., RAGHAVAN, V. V. ELAYAVALLI, V. K. (1991) Genetic algorithm for clustering with an ordered representation. Fourth International Conference on Genetic Algorithms. San Mateo, CA, Morgan Kaufman.

BLASS, W. E. (1963) Rotation-Vibration Spectra of Axially Symmetric Molecules with Localized Perturbations. Department of Physics and Astronomy. East Lansing, MI, Michigan State University.

BLASS, W. E. (1976) Data Acquisition, Reduction, and Analysis in Infrared and Optical Spectroscopy. Applied Spectroscopy Reviews, 11, 57-123.

BLASS, W. E. NIELSEN, A. H. (1974) IN WILLIAMS, D. (Ed.) Method of Experimental Physics 2nd ed. New York, Academic.

BOCK, H. H. (1985) On some significance tests in cluster analysis. Journal of Classification, 2, 77-108.

BOCK, H. H. (1996) Probability models in partitional cluster analysis. Computational Statistics and Data Analysis, 23.

BOYCE, D. E., FARHI, A. WEISCHEDEL, R. (1974) Optimal Subset Selection: Multiple Regression, Interdependence, and Optimal Network Algorithms, New York, Springer-Verlag.

BOYD, J. W. (1963) Analysis of Vibration-Rotation Spectra of Axially Symmetric Molecules, with Applications to CD3I. Department of Physics and Astronomy. East Lansing, MI, Michigan State University.

BOZDOGAN, H. (1981) Multi-Sample Cluster Analysis and Approaches to Validity Studies in Clustering Individuals. Department of Mathematics. Chicago, IL, University of Illinois at Chicago.

BOZDOGAN, H. (1983) Determining the Number of Component Clusters in the Standard Multivariate Normal Mixture Model Using Model-Selection Criteria. Chicago, IL, University of Illinois at Chicago, Quantitative Methods Department.

BOZDOGAN, H. (1987) Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions. Psychometrica, 52, 345-370.

BOZDOGAN, H. (1988) ICOMP: A New Model Selection Criterion. IN BOCK, H. H. (Ed.) Classification and Related Methods of Data Analysis. Amsterdam, North-Holland.

BOZDOGAN, H. (1990a) On the Information-Based Measure of Covariance Complexity and its Application to the Evaluation of Multivariate Linear Models. Communications in Statistics (Theory and Methods), 19, 221-278.

BOZDOGAN, H. (1990b) Multisample Cluster Analysis of the Common Principal Component Model in K Groups Using An Entropic Statistical Complexity Criterion. International Symposium on Theory and Practice of Classification. Puchino, Soviet Union.

BOZDOGAN, H. (1992) Choosing the Number of Component Clusters in the Mixture-Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix. IN OPITZ, O., LAUSEN, B. KLAR, R. (Eds.) Studies in Classification, Data Analysis, and Knowledge Organization. Heidelberg, Germany, Springer-Verlag.

BOZDOGAN, H. (1993) Choosing the number of component clusters in the mixture model using a new informational complexity criterion of the inverse fisher information matrix. IN OPITZ, O., LAUSEN, B. KLAR, R. (Eds.) Information and Classification. Springer-Verlag.

BOZDOGAN, H. (1994) Mixture-Model Cluster Analysis using a Model Selection Criteria and a New Informational Measure of Complexity. IN BOZDOGAN, H. (Ed.) First US/Japan Conference on the Frontiers of Statistical Modeling: An Informational Approach. Kluwer Academic Publishers.

BOZDOGAN, H. (2000) Akaike's information criterion and recent developments in information complexity. Journal of Mathematical Psychology, 44, 62-91.

BOZDOGAN, H. (2001) Statistics 563 Class Notes. University of Tennessee.

BOZDOGAN, H. (2004) Statistical data mining and knowledge discovery Boca Raton, FL, Chapman Hall/CRC.

BROWNLEE, K. A. (1965) Statistical Theory and Methodology in Science and Engineering, New Yory, John Wiley Sons, Inc.

CALINSKI, T. HARABASZ, J. (1974) A Dendrite Method for Cluster Analysis. Communications in Statistics, 3, 1-27.

CAMERON, A. C. TRIVEDI, P. K. (1998) Regression analysis of count data, Cambridge, United Kingdom, Cambridge University Press.

COUSINS, R. D. (1995) Why isn't every physicist a Bayesian? American Journal of Physics, 63, 398-410.

DARLING, B. T. DENNISON, D. M. (1940) Physical Reviews, 57, 128.

DE JONG, K. (1990) Genetic algorithm-based learning, San Mateo, CA, Morgan Kaufmann.

DEMPSTER, A., LAIRD, N. M. RUBIN, D. B. (1977) Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, 29, 1-38.

DRAPER, N. R. SMITH, H. (1966) Applied Regression Analysis, New York, John Wiley Sons, Inc.

EFROYMSON, M. A. (1960) Multiple Regression Analysis. IN RALSTON, A. WLIF, H. S. (Eds.) Mathematical Methods for Digital Computers. New York, John Wiley.

ENGELMAN, L. HARTIGAN, J. A. (1969) Percentage points of a test for clusters. Journal of the American Statistical Association, 64, 1647-1648.

FORREST, S. (1993) Genetic Algorithms: principles of natural selection applied to computation. Science, 261, 872-878.

GOLDBERG, D. E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, New York, Addison-Wesley. GORDON, A. D. (1999) Classification: Methods for the Exploratory Analysis of Multivariate Data, New York, Chapman and Hall.

GUELACHVILI, G., KOIVUSAARI, M. ANTTILA, R. (1984) The ν_4 Band of CD_3I with Perturbations: A High-Resolution Infrared Study. Journal of Molecular Spectroscopy, 108, 99-118.

HAFFORD, J. A. (1972) Development of a Stepwise Regression Analysis System for Vibration-Rotation Spectra of Axially Symmetric Molecules. Department of Physics and Astronomy. Knoxville, TN, University of Tennessee.

HAHN, G. J. (1977) Fitting Regression Models with No Intercept Term. Journal of Quality Technology, 9, 56-61.

HARTIGAN, J. A. (1975) Clustering Algorithms, New York, Wiley.

HENNA, J. (1985) On Estimating of the Number of Constituents of a Finite Mixture of Continuous Distributions. Annals of the Institute of Statistical Mathematics, Part A, 37, 235-240.

HENNA, J. (1986) An Application of a Mixture Method to Classification. Journal of Japan Statistical Society, 16, 133-143.

HOCKING, R. R. (1976) The analysis and selection variables in linear regression. Biometrics, 32, 1044.

HOFFMAN, F., HARGROVE, W. H., ERICKSON, D. J. OGLESBY, R. J. (2005) Using Clustered Climate Regimes to Analyze and Compare Predictions from Fully Coupled General Circulation Models. Earth Interactions, 9, 1-27.

HOLLAND, J. (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press.

HOLLAND, J. (1992) Genetic Algorithms. Scientific American, 66-72.

HUBBLE, E. (1929) A Relation Between Distance and Radial Velocity Among Extra-Galactic Nebulae From the Proceedings of the National Academy of Sciences.

HUMPHREYS, R. M., KARYPIS, G., HASAN, M., KRIESSLER, J. ODEWAHN, S. C. (2001) Experiments in Automating the Morphological Classification of Galaxies. 199th Meeting of American Astronomical Society. Washington, DC.

JAIN, A. K. DUBES, R. C. (1989) Algorithms for Clustering Data, Englewood Cliff, NJ, Prentice-Hall.

JARVIS, R. A. PATRICK, E. A. (1973) Clustering Using a Similarity Measure Based on Shared Near Neighbors. IEEE Transactions on Computers, C22, 1025-1034.

JEFFERYS, W. H. BERGER, J. O. (1992) Ockham's razor and Bayesian analysis (statistical theory for systems evaluation) American Scientist, 80, 64-72.

JONES, D. R. BELTRAMO, M. A. (1991) Solving Partitioning Problems with Genetic Algorithms. IN BELEW, K. R. BOOKER, L. B. (Eds.) Fourth International Conference on Genetic Algorithms. San Mateo, CA, Morgan Kaufmann.

KAUFMAN, L. ROUSSEEUW, P. J. (1990) Finding Groups in Data. An Introduction to Cluster Analysis, New York, Wiley.

KENDALL, M. G. (1957) A Course on Multivariate Analysis, London, Griffin and Co.

KENDALL, M. G. STUART, A. (1966) Advanced Theory of Statistics, London, Griffin and Co.

KOZA, J. R. (1992) Genetic programming : on the programming of computers by means of natural selection Cambridge, MA, MIT Press.

KRISHNA, K. MURTY, M. (1999) Genetic K-means algorithm. IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, 29, 433-439.

KULLBACK, S. LEIBLER, R. (1951) On information and sufficiency. Annals of Mathematical Statistics, 22, 79-86.

KURLAT, H., KURLAT, M. BLASS, W. E. (1971) Simultaneous Least Squares Analysis of $\nu_2 + \nu_4$, $\nu_4 + \nu_5$, and $2\nu_4$ of CD_3I. Journal of Molecular Spectroscopy, 38, 197-213.

KURLAT, H. K. (1970) Interpretation of Several Vibration Rotation Bands of CD3I. Department of Physics and Astronomy. Knoxville, TN, University of Tenness.

KURLAT, M. (1969) An Analysis System for Vibration-Rotation Spectra of Axially Symmetric Molecules. Department of Physics and Astronomy. Knoxville, TN, University of Tennessee.

LAWRENCE, A. (1987) Classification of Active Galaxies and the Prospect of a Unified Phenomenology. Publications of the Astronomical Society of the Pacific, 99, 309.

LUO, A., ZHANG, Y., ZHANG, J. ZHAO, Y. (2004) Mining the LAMOST Spectra Archive. Astronomical Data Analysis II. Proc. of SPIE. MACQUEEN, J. B. (1967) Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, CA.

MANN, D. E., THRUSH, B. A., LIDE, D. R. J., BALL, J. J. ACQUISTA, N. (1961) Spectroscopy of fluorine flames. I. Hydrogen-fluorine flame and the vibration-rotation emission spectrum of HF. Journal of Chemical Physics, 34.

MANTEL, N. (1970) Why stepdown procedures in variable selection. Technometrics, 12, 591-612.

MAO, J. JAIN, A. K. (1996) A self-organizing network for Hyperellipsoidal Clustering (HEC). IEEE Transactions on Neural Networks, 7, 16-29.

MARDIA, K. V., KENT, J. T. BIBBY, J. M. (1979) Multivariate Analysis, New York, Academic Press.

MARONNA, K. V. JACOVKIS, P. M. (1974) Multivariate Clustering Procedures with Variable Metrics. Biometrics, 27, 501-514.

MARRIOTT, F. H. C. (1971) Practical Problems in a Method of Cluster Analysis. Biometrics, 27, 501-514.

MARTINEZ, W. L. MARTINEZ, A. R. (2002) Computational Statistics Handbook with MATLAB, Boca Raton, FL, CRC Press.

MATUSITA, K. OHSUMI, N. (1981) Evaluation Procedure of Some Clustering Techniques.

MCLACHLAN, G. J. KRISHNAN, T. (1997) The EM Algorithm and Extensions, New York, John Wiley Sons.

MILLER, A. J. (1996) The Convergence of Efroymson's Stepwise Regression Algorithm. The American Statistician, 50, 180-182.

MITCHELL, M. (1998) An Introduction to Genetic Algorithms, Cambridge, MA, MIT Press.

MONTGOMERY, D. C., PECK, E. A. VINING, G. G. (2001) Introduction to Linear Regression Analysis, New York, John Wiley Sons, Inc.

MOSES, L. E. (1986) Think and Explain with Statistics, Reading, MA, Addison-Wesley.

MURTAGH, F. HECK, A. (1987) Multivariate Data Analysis, Reidel, Dordrecht.

NETER, J., WASSERMAN, W. KUTNER, M. H. (1990) Applied Linear Statistical Models, Regression, Analysis of Variance, and Experimental Designs, Homewood, IL, Irwin.

ODEWAHN, S. C., STOCKWELL, E. B., PENNINGTON, R. L., HUMPHREYS, R. M. ZUMACH, W. A. (1992) Automated star/galaxy discrimination with neural networks. Astronomical Journal, 103, 318.

PETERS, B. C. WALKER, H. F. (1978) An Iterative Procedure for Obtaining Maximum-Likelihood Esimates of the Parameters for a Mixture of Normal Distributions. SIAM Journal of Applied Mathematics, 35, 362-378.

RECHENBERG, I. (1973) Evolutionsstrategie, Stuttgart. RICE, J. A. (1995) Mathematical Statistics and Data Analysis, Belmont, CA, Duxbury Press.

RISSANEN, J. (1978) Modeling by shortest data description. Automatica, 14, 465-471.

RISSANEN, J. (1986) Stochastic complexity and modeling. Ann. Statist., 14, 1080-1100.

ROEDER, K. WASSERMAN, L. (1997) Practical Bayesian Density Estimation Using Mixture of Normals. Journal of the American Statistical Association, 92, 894-902.

RUDOLPH, G. (1997) Convergence rates of evolutionary algorithms for a class of convex objective functions Control and Cybernetics, 26, 375-390.

SCHWARTZ, G. (1978) Estimating the dimension of a model. Ann. Statist., 6, 461-464.

SCOTT, A. J. SYMONS, M. J. (1971) Clustering methods based on likelihood ratio criteria. Biometrics, 27, 387–397.

SHANNON, C. E. (1948) A mathematical theory of communication. Bell Systems Technology Journal, 27, 379-423.

SOKAL, R. R. ROHLF, F. J. (1981) Biometry, New York, W.H. Freeman and Company.

SOUBIRAN, C. (1993) Kinematics of the Galaxy's stellar populations from a proper motion survey. Astronomy and Astrophysics, 274, 181-188.

STORRIE-LOMBARDI, M. C., LAHAV, O., SODRE, L., JR. STORRIE-LOMBARDI, L. J. (1992) Morphological Classification of Galaxies by Artificial Neural Networks. Monthly Notices of the Royal Astronomical Society, 259.

SUMIDA, B. H., HOUSTON, A. I., MCNAMARA, J. M. HAMILTON, W. D. (1990) Genetic algorithms and evolution. Journal of Theoretical Biology, 147, 59-84.

SZLAY, A. GRAY, J. (2001) The World-Wide Telescope. Science, 293, 2037-2040.

SZLAY, A., GRAY, J. VANDENBERG, J. (2002) Petabyte Scale Data Mining: Dream or Reality? Redmond, WA, Microsoft Research.

VAN EMDEN, M. H. (1971) An Analysis of Complexity, Amsterdam.

VAPNIK, V. N. (1995) The Nature of Statistical Reasoning, New York, Springer.

WALKER, E. (2002) Statistics 572 Class Notes. University of Tennessee.

WANG, S., MA, F., SHI, W. XIA, S. (1997) The Hyperellipsoidal Clustering Using Genetic Algorithm. IEEE International Conference on Intelligent Processing Systems. Beijing, China.

WANG, S. XIA, S. (1997) Comments on "A Self-organizing Network for Hyperellisoidal Clustering (HEC)". IEEE Transactions on Neural Networks, 8, 1561-1562.

WILKINSON, L. (1989) SYSTAT: The System for Statistics, Evanston, IL, SYSTAT.

WONG, M. A. (1982) A Hybrid Clustering Method for Identifying High-Density Clusters. Journal of the American Statistical Association, 77, 841-847.

WOZNIAK, P. R. (2001) Classification of ROTSE Variable Stars using Machine Learning. 199th Meeting of American Astronomical Society. Washington, DC. WRIGHT, S. (1931) Evolution in Mendelian Populations. Genetics, 16, 97-159.

ZHANG, Y. ZHAO, Y. (2003) Classification in Multidimensional Parameter Space: Methods and Examples. Publications of the Astronomical Society of the Pacific, 115, 1006-1018.

ZHANG, Y. X. ZHAO, Y. H. (2004) Automated clustering algorithms for classification of astronomical objects. Astronomy and Astrophysics, 422, 1113.

Vita

James Eric Wicker was born in 1975 and grew up in Arcadia, Florida. As an undergraduate, he attended New College of the University of South Florida (now called New College Florida) in Sarasota. He graduated in 1997 with a major in Physics. He attended the University of Tennessee, Knoxville, for graduate school where he was a double major in Physics and Statistics. He won the UT Physics Department's 2003 Outstanding Graduate Teaching Assistant Award. That year, he also completed a Master of Science degree in Statistics. In 2004, he was accepted into the first NSF Summer Institute in China exchange program where he worked at the National Astronomical Observatory of China in Beijing. He was subsequently invited to give a presentation about his experiences during the exchange program at the US-China Joint Commission Meeting hosted by the US State Department in Washington, D.C. In 2005, he was awarded a Chancellor's Citation for Extraordinary Professional Promise.