Virginia Commonwealth University
School of Medicine

This is to certify that the dissertation prepared by Eric Scott Harvey entitled "Normal Mixture Models for Gene Cluster Identification in Two Dimensional Microarray Data" has been approved by his committee as satisfactory completion of the dissertation requirement for the degree of Doctor of Philosophy.

Viswanathan Ramakrishnan, Ph.D., Director of Dissertation

Kellie J. Archer

Kellie Archer, Ph.D., School of Medicine

Lindon Eaves

Lindon Eaves, D.Sc., Ph.D., School of Medicine

R. K. Elswick, Ph.D., School of Medicine

Brad Windle, Ph.D., School of Medicine

Walter H. Carter, Jr., Ph.D., Department Chair

H H Newsome

H. H. Newsome, Jr., M.D., Dean, School of Medicine

F. Douglas Boudinot, Ph.D., Dean, School of Graduate Studies

June 25, 2003

Date

Normal Mixture Models for Gene Cluster Identification in Two Dimensional Microarray Data

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

By

Eric Scott Harvey

Bachelor of Science in Mathematics, 1999
Radford University
Radford, VA

Master of Science in Information Systems, 2001
Strayer University
Midlothian, VA

Director: Dr. Viswanathan Ramakrishnan

Associate Professor
Department of Biostatistics

Virginia Commonwealth University
Richmond, Virginia
August, 2003

Acknowledgement

Many people have contributed toward my achievements, and I would like to thank them. First of all, I greatly appreciate the love and support that my wife Paige has shown me throughout the creation of this dissertation. I would not have made it through without her.

My parents have continually encouraged me throughout my academic career. I thank them for their continued support and belief in my ability to succeed. Thanks also to my Aunt Nancy and Uncle John and to my sister Amy for helping me to stay organized and for proofreading something that is not easy to read. I am grateful to them and to all of my friends and family.

I would also like to thank my dissertation director, Dr. Viswanathan Ramakrishnan. His guidance has been invaluable and my knowledge of biostatistics has grown greatly because of him. Thanks also to my committee members, including Dr. Kellie Archer for her careful proofreading and statistical insights, Dr. Lindon Eaves for helping me to not lose sight of the "big picture", Dr. R. K. Elswick for his constant encouragement and keeping me on track to graduate, and Dr. Brad Windle for helping with the relevant biology and providing the data sets.

A special thanks goes to Dr. J. Susan Milton for teaching my first statistics course. She was one of the best teachers that I have had. I have been hooked on statistics ever since.

This research was partially funded by grant number T32MH20030 from the National Institutes of Health (NIH) through an Institutional National Research Service Award (NRSA). Dr. Michael Neale of the Virginia Institute for Psychiatric and Behavioral Genetics (VIPBG) at Virginia Commonwealth University is the principal investigator. I would like to thank Dr. Neale for his help with understanding structural equation models and increasing my knowledge of behavioral genetics.

Finally, to my soon to be born daughter Sarah, I'm finally done and ready to be sleep deprived for a different reason!

Table of Contents

Page

List of Tables

Table Page

List of Figures

Figure Page

Abstract

NORMAL MIXTURE MODELS FOR GENE CLUSTER IDENTIFICATION IN TWO DIMENSIONAL MICROARRAY DATA

By Eric Scott Harvey

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2003

Director: Dr. Viswanathan Ramakrishnan
Associate Professor
Department of Biostatistics

This dissertation focuses on methodology specific to microarray data analyses that organize the data in preliminary steps and proposes a cluster analysis method which improves the interpretability of the cluster results. Cluster analysis of microarray data allows samples with similar gene expression values to be discovered and may serve as a useful diagnostic tool. Since microarray data is inherently noisy, data preprocessing steps including smoothing and filtering are discussed. Comparing the results of different clustering methods is complicated by the arbitrariness of the cluster labels. Methods for re-labeling clusters to assess the agreement between the results of different clustering techniques are proposed.

Microarray data involve large numbers of observations and generally present as arrays of light intensity values reflecting the degree of activity of the genes. These measurements are often two dimensional in nature since each is associated with an individual sample (cell line) and gene. The usual hierarchical clustering techniques do not easily adapt to this type of problem. These techniques allow only one dimension of the data to be clustered at a time and lose information due to the collapsing of the data in the opposite dimension. A novel clustering technique based on normal mixture distribution models is developed. This method clusters observations that arise from the same normal distribution and allows the data to be simultaneously clustered in two dimensions. The model is fitted using the Expectation/Maximization (EM) algorithm. For every cluster, the posterior probability that an observation belongs to that cluster is calculated. These probabilities allow the analyst to control the cluster assignments, including the use of overlapping clusters.

A user friendly program, 2-DCluster, was written to support these methods. This program was written for Microsoft Windows 2000 and XP systems and supports one and two dimensional clustering. The program and sample applications are available at http://etd.vcu.edu. An electronic copy of this dissertation is available at the same address.

Chapter 1

Introduction

The goal of cluster analysis is to identify homogenous subgroups within complex datasets. Cluster analysis attempts to discover natural groupings based on some measure of similarity (or dissimilarity) between objects. If the data can be validly summarized by grouping objects, then the group labels may provide enough information to describe patterns and similarities in the data (Everitt *et al*., 2001). If no patterns or similarities are present in the data, only one cluster containing all of the observations should exist. However, hierarchical clustering methods (discussed in Chapter 2) always produce cluster results regardless of whether or not these patterns are present. Clustering is different from classification in the sense that the number of groups and the group labels are known *a priori* in classification, whereas cluster analysis users must frequently make assumptions concerning the number of groups or the group structure (Johnson and Wichern, 1998).

There are many cluster analysis techniques available. When only two variables are involved, examining histograms offers some idea of where clusters lie. Graphical methods are also useful for visualizing three dimensional data. However, higher dimensional data quickly become too complex to interpret visually. As Carl Sagan (1986) wrote, "Humans are good at discerning subtle patterns that are really there, but equally so at imagining them when they are altogether absent." This statement is particularly relevant when visually examining clusters in multivariate datasets.

Cluster analysis offers formal statistical tools to help to assign observations to clusters. This dissertation develops tools specific to microarray data analysis that organize the data in preliminary steps to fine tune cluster analysis. A new parametric method for clustering simultaneously in two dimensions is proposed.

1.1 Microarray Technology

Microarrays allow researchers to measure the expression levels of a large number of genes simultaneously. Only about 40 percent of genes on average are expressed at a given time (Lockhart, 2002). The two most commonly used microarray technologies are the custom spotted two-color complementary DNA (cDNA) microarray and the oligonucleotide microarray (*e.g.* Affymetrix gene chips). The primary difference between these designs is that the cDNA approach uses a single long stretch of DNA for each gene while the Affymetrix approach uses several short oligonucleotides to probe for each gene. The cDNA technology measures the relative gene abundance from two samples while the Affymetrix technology measures the absolute gene abundance for a single sample.

For the two-color cDNA microarray, the DNA from thousands of genes is spotted onto a small glass slide in a regular pattern. Each spot or probe interrogates for a specific gene. This approach was pioneered by Pat Brown's laboratory at Stanford University (Schena *et al.*, 1995; Brown and Botstein, 1999). Probes are generated by amplifying genomic DNA with gene specific primers. The probes are spotted onto the slide automatically by a robot. Messenger RNA (mRNA) from the samples is purified and reverse transcribed to cDNA with fluorescent labeled nucleotides. If two samples are used (*e.g.* control and treatment), they are labeled separately with the fluorescent dyes Cyanine-3 (Cy3) and Cyanine-5 (Cy5), which emit light in different spectrums (Mujumdar *et al.*, 1993). The spectrums are assigned the colors green (Cy3) and red (Cy5) for convenience. The labeled cDNA is mixed in equal amounts and hybridized to the array. Unbound cDNA is washed away and the array is scanned twice with a laser, generating one red and one green image. Once the images are overlaid, spots hybridized with equal amounts of control and treatment cDNA are yellow, while spots for genes that are differentially expressed are different shades of red or green. The cDNA microarray design is illustrated in Figure 1.1.

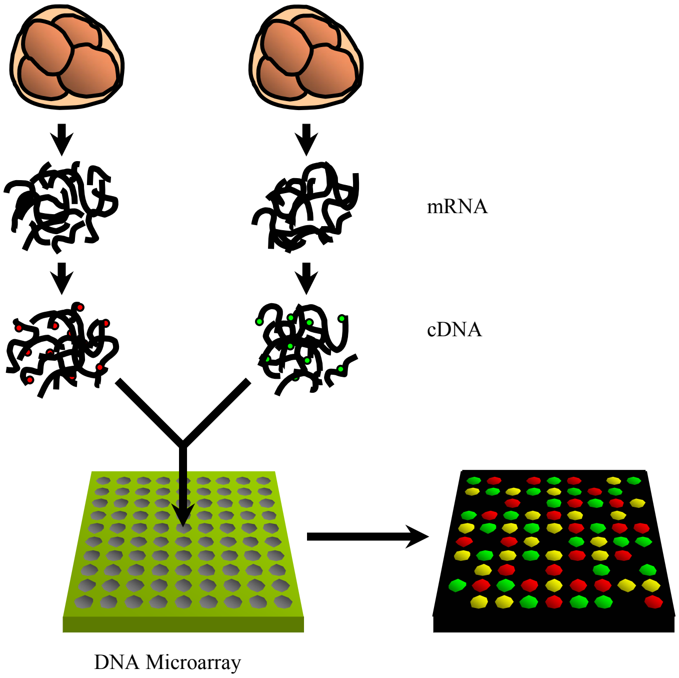


Figure 1.1: cDNA Microarray Design

Various image analysis techniques are employed to identify the red and green intensities in the spots along with the surrounding background. Since the spot size and hybridization properties change for different nucleotide sequences, the measured fluorescence intensity cannot be translated to an absolute level of mRNA.

The ratio between the amount of gene specific mRNA in the two samples is called a fold difference. Historically, a fold difference of two or more was often interpreted as evidence that the gene is differentially expressed.

Due to the variability in spot size and other variations in the printing process, normalization techniques are often employed to aid in comparisons across multiple arrays or experiments run with different samples or conditions. Normalization is intended to compensate for systematic errors not due to biology, while repeated measurements help to control for random errors. When normalizing data, researchers commonly subtract the background intensity from the foreground intensity before transforming, say by a logarithm. The base 2 logarithm is often employed because the intensities are measured on a 16 bit scale. One must be careful not to over-normalize the data. See Quackenbush (2001) or Quackenbush (2002) for a good discussion on microarray data normalization.

The Affymetrix oligonucleotide microarray design was perfected by David Lockhart *et al.* (1996). Affymetrix microarrays are packaged in an easy to handle chip as shown in Figure 1.2.

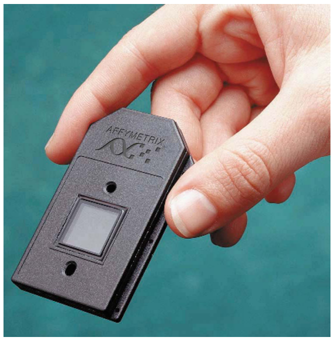


Figure 1.2: DNA Chip Illustration

Oligonucleotides are placed on glass slides using combinatorial chemistry combined with a photolithographic process. The 25-mer oligonucleotides are used as probes for specific genes. Each is located in a specific area on the array called a probe cell. The probe cells can contain millions of copies of a given oligonucleotide. Tens to hundreds of thousands of different oligonucleotide probes are synthesized on each array.

Probe arrays are manufactured in a series of cycles. Initially, a glass substrate is coated with linkers containing photolabile protecting groups. As shown in Figure 1.3, a mask is applied that exposes selected portions of the probe array to ultraviolet light. The light removes the photolabile protecting groups which enables the addition of nucleotide material only at the previously exposed sites. This process is repeated using different masks and illumination cycles. By repeating these steps, a specific set of oligonucleotide probes is synthesized with each probe type in a known location. The completed probe arrays are packaged into cartridges like the one shown in Figure 1.2.

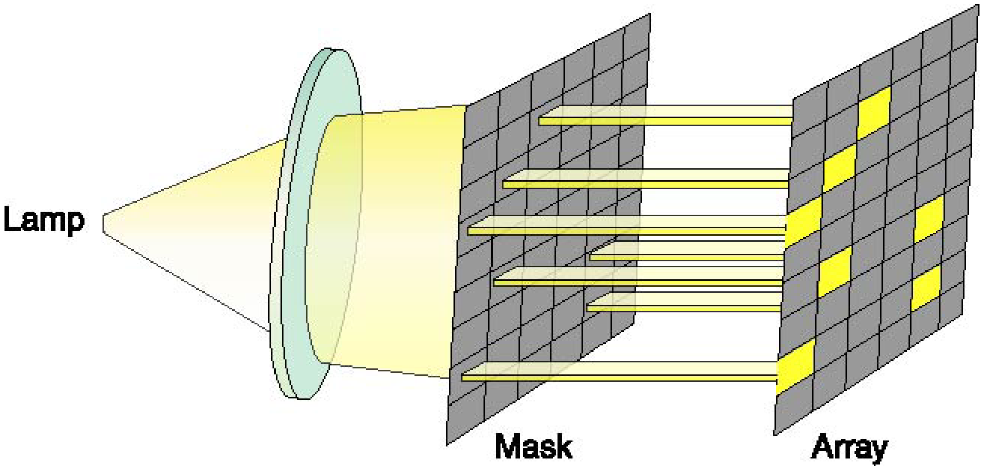


Figure 1.3: Photolithographic DNA Synthesis

Lockhart (2002) lists the following steps in synthesizing DNA for an entire DNA chip.

1. Spatially-specific illumination
2. Illuminated oligonucleotides de-protected
3. Coupling of protected nucleotide
4. Next illumination pattern applied
5. Next base coupling step
6. Process repeated to build entire chip

These steps are illustrated in Figure 1.4.

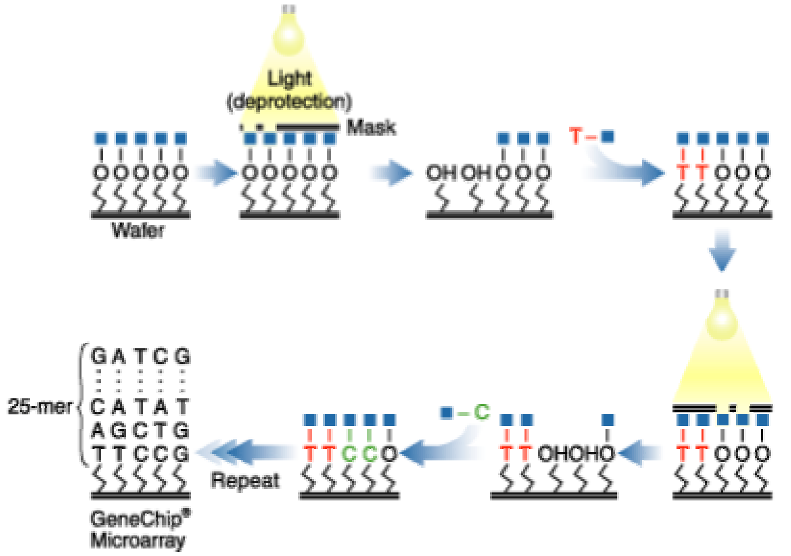


Figure 1.4: DNA Synthesis Process

In the Affymetrix technology, individual oligonucleotides are represented by a unique sequence of 25 base pairs. This sequence is called a 25-mer. Each gene is usually represented by eleven 25-mers on the Hu-133A and Hu-133B chips. Any set of 25-mers can be made in fewer than 100 synthesis steps. Probes are made by creating a 25-mer which is complementary to the reference sequence. A perfect match probe, or PM, is a 25-mer which is the exact complement of the reference sequence. A mismatch probe, or MM, is a 25-mer which is the same as the PM except for a base change in the middle (13th) base. The purpose of the MM probe design is to measure non-specific binding and background noise. Most expression measures are based on differences of PM and MM (PM-MM). A probe pair refers to a (PM, MM) pairing. Eleven probe pairs make up the typical probe set. The individual probe cells are square shaped. When the microarray is “read”, the resulting image contains about 100 pixels per probe cell. The eleven probe cell PM and MM intensities are combined to form an expression level for a probe set. Bolstad *et al.* (2003) propose a different normalization method which attempts to remove the obscuring variation.

Both the cDNA and Affymetrix microarray designs have merit. The long DNA strands in the cDNA design are more specific than oligonucleotides. However, Affymetrix chips have many oligonucleotides per gene compared to just a few spots per gene on cDNA arrays. Affymetrix arrays have a mismatched control for every oligonucleotide. These mismatches may or may not be informative. Affymetrix software uses robust weighting techniques to combine the signals of the PM and MM oligonucleotide pairs into a probe set expression summary. However, much of this information is proprietary. The probe sequences only recently became publicly available (Lockhart, 2002). Affymetrix experiments have a single color readout versus the two colors used in cDNA experiments. The cDNA experiments are cheaper to run and the equipment is more readily available. Affymetrix experiments are arguably more reliable but the equipment and gene chips cost more (although the costs are constantly falling). Finally, Affymetrix microarrays are much denser and can handle many more genes simultaneously than cDNA microarrays.

1.2 Computational Complexity

In most cluster analysis applications, the user knows enough about the problem to be able to distinguish “good” clusters from “bad” clusters. One option is to list all of the possible groupings and to choose the best clusters based on some criteria (such as minimizing variability) for further study. However, this approach becomes computationally infeasible when the number of genes or groups is large.

For example, suppose that a researcher is interested in studying smoking in the United States population and that one demographic variable is chosen from a data set containing 25 smokers. There is only one way to form a cluster of size one. There are 16,777,215 ways to partition the subjects into two clusters (of varying sizes). There are $1.41x10^{11}$ ways to partition the patients into three clusters (of varying sizes). In general, the number of ways of sorting *n* objects into *k* nonempty groups is a Stirling number of the second kind given by Johnson and Wichern (1998):

$$NC = \frac{1}{k!}\sum_{j=0}^{k}\left[(-1)^{k-j}\binom{k}{j}j^{n}\right]. \tag{1.1}$$

The SAS code for these calculations is given in Appendix 1.1. Clearly, listing all of the possible clusters is not practical even for this small example. The complexity quickly increases as the number of variables and subjects increases. Cluster analysis does not seek to find the absolute best cluster assignment by enumerating all NC possibilities. It only strives to find cluster assignments that are useful in practice by user defined criteria for homogeneity of units within clusters.

1.3 Clustering

The goal of clustering is to group "similar" observations together. In the usual clustering application, each object belongs to a single cluster and the complete set of clusters contains all of the objects. However, in some scientific applications, allowing overlapping clusters is useful. For example, a non-overlapping structure for clusters may not be appropriate for microarray data analysis, as there is no reason to assume that cancer causing genes can only be active in one type of cancer.

Everitt *et al.* (2001) give the usual form of the data for cluster analysis applications as an $n\ x\ p$ matrix, **X**, containing the observations describing each object to be clustered.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \tag{1.2}$$

The entry x_{ij} in **X** gives the value of the j^{th} variable for object *i*. The variables may be a mixture of continuous, ordinal, or nominal types and could also be missing. Most clustering methods reduce **X** to a $n\ x\ n$ symmetric matrix of distances or similarities (see Chapter 2 for more details). Similarity measures scaled to fall between 0 and 1 convert to distance measures by subtracting them from 1. Some clustering algorithms require the number of clusters to be specified in advance, which is often unknown and thus can be problematic.

Suppose that a researcher wishes to look for groups of people having a similar IQ in a sample of 100 individuals. Since traditional clustering is based on a distance (or similarity) measurement, this analysis is based on a 100x100 symmetric matrix of distances having $\frac{n(n+1)}{2} = \frac{100x101}{2} = 5050$ unique entries. The result of running the cluster analysis is clusters of individuals having "similar" IQs. Any clusters found in this crude analysis could be further examined to see if there are demographic or socioeconomic commonalities among the groups.

There are certain procedures common to most clustering algorithms. Cowgill (1993) lists the following general steps that are involved in a cluster analysis.

1. The objects to be clustered must be selected. For convenience in data collection, analyses are typically performed on a population sample.
2. The variables to be used in the cluster analysis must be selected. The variables must contain enough information to permit the clustering of the objects. The proportion of relevant information to random error (noise) should be kept to a minimum.
3. The researcher must decide whether or not to standardize the raw data. Additionally, the decision must be made whether to categorize continuous data into finite groups.
4. A distance (or similarity) measure must be chosen.
5. A clustering technique must be chosen.
6. The number of clusters must be ascertained.
7. The researcher must interpret the meaning of the clusters in terms of the research objectives.

1.4 Historical Development of Cluster Analysis

Clustering techniques were first discussed in the social science literature in 1930 (Blashfield and Aldenderfer, 1978). Early interest in clustering biological organisms was sparked by the publication of the classic text *Principles of Numerical Taxonomy* by Sokal and Sneath (1963). However, clustering techniques have only found widespread application in the past 25 years. Many of the clustering techniques used today were developed in the 1950's and 1960's and are now feasible for a much wider user base due to the rapid growth of computer speed. A few of these techniques are single linkage (Sneath, 1957), average linkage (Sokal and Michener, 1958), complete linkage (Sorensen, 1948), and Ward's method (Ward, 1963). These techniques are discussed more completely in Chapter 2. Chapter 4 introduces mixture model based clustering. Cluster analysis is being used in diverse fields such as agriculture, archaeology, astronomy, business, and psychiatry. Data mining is one of the fastest growing approaches for pattern recognition and makes use of a wide variety of clustering techniques.

1.5 Multidimensional Clustering

Current clustering techniques require the selection of a dimension to cluster across. For example, suppose that a researcher wishes to analyze data from a microarray experiment. Let the columns of the data represent samples and the rows represent genes. The goal is to find clusters of genes which have similar gene expression patterns in a given cluster of samples. Many studies of this type cluster across samples which results in clusters of "similar" samples.

The researcher's basic question regards discovering relationships between genes and samples. Choosing one dimension (*e.g.* the samples) of the data to cluster across effectively ignores data regarding the relationships between the genes and instead focuses only on the differences in aggregate gene expression between the samples. Thus, information is lost due to this artificial choice of a clustering dimension.

1.6 Clustering Microarray Data

An established cDNA microarray data set from the National Cancer Institute is made up of a two dimensional array having 6,167 genes across one axis and 60 cell lines across the other axis (Ross *et al.*, 2000). The observations in the grid are gene expression values. The cluster analysis of microarray data represents a huge computational challenge due to the large number of observations.

As shown in Figure 1.5, analyzing and interpreting a microarray experiment involves several steps.

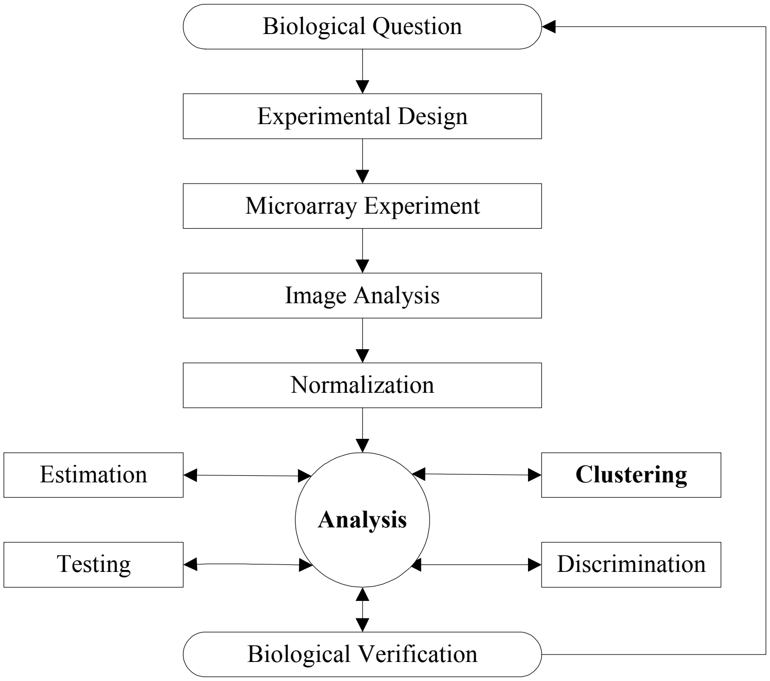


Figure 1.5: Steps in the Analysis of Microarray Data

This dissertation focuses on the analysis step (bolded in Figure 1.5) and applies cluster analysis techniques. We assume that the gene expression values reported in the experiments were appropriately obtained. Cluster analyses are one type of analysis that may be performed on microarray data. It is noted, however, that the image analysis and normalization procedures applied may greatly influence the results. As always, good laboratory practices including a complete protocol can help to control noise and to improve the interpretability of the results. Any conclusions drawn from the analysis stage should be subjected to biological verification and interpretation.

Brazma *et al.* (2001) proposed guidelines for presenting and exchanging microarray data, known as the Minimum Information About a Microarray Experiment (MIAME) standard. The goal of these guidelines is to outline the minimum information required to interpret unambiguously and potentially reproduce and verify an array based gene expression monitoring experiment. The MIAME standards are widely used and continually updated. The current version is found at http://www.mged.org/Workgroups/MIAME/miame_1.1.html.

Medical applications of microarray data analysis may seek to identify genes involved in disease by comparing gene expression values between tissues of healthy and diseased individuals. This is often accomplished by supervised learning techniques for class comparison and class prediction. Alternatively, unsupervised learning methods are useful for class discovery. Moreover, patterns of genes specifically induced in pathological tissues may be identified using clustering techniques. Finding genes that are common to specific groups of tumors may prove useful. Such findings could offer medical researchers a starting place in their quest to improve the reliability of cancer diagnosis and treatment effectiveness. The possibility of gene targeted treatment requires one to more accurately understand the underlying genetic and environmental factors which contribute to the development of cancer.

Cluster analysis is one tool in a growing arsenal of research weapons for better understanding these relationships.

Cluster analysis of microarrays lends itself to other applications as well. Using microarrays in functional genomic studies offers clues to discovering gene function through the examination of gene expression patterns. Microarrays are also used to study treatment effects on metabolic and signaling pathways. Eventually, researchers hope to be able to determine the structure of regulatory gene networks by analyzing expression data. Finally, microarrays are used in comparative genomic studies in the hopes of identifying genetic differences between closely related species.

1.7 Research Focus

This dissertation focuses on developing tools specific to microarray data analysis that organize the data in preliminary steps to fine tune the cluster analysis and improve the interpretability of the cluster results. One dimensional non-parametric clustering techniques are reviewed. Issues such as smoothing and gene filtering are examined. Methods for assessing the agreement between the results of different clustering techniques are addressed. One dimensional parametric clustering techniques are discussed. A parametric algorithm that examines both dimensions of the data simultaneously in order to establish clusters is developed. A statistical framework is developed for this algorithm. A user friendly program, 2-DCluster, implements the algorithm. Finally, suggestions for future research in this area are given along with a brief review of other methods currently in vogue and a summary of the results contained in this dissertation.

Chapter 2

Smoothing, Filtering, and Non-Parametric Clustering

2.1 Introduction

For microarray data, clustering methods are used to identify genes that have similar expression patterns with respect to a sample. In this dissertation, a brief account of the common clustering methods and the issues related to their application is given. One of the primary requirements for any clustering method is the specification of a "distance" measure or a "similarity" measure. Different distance and similarity measures are defined and discussed. Several unidimensional clustering algorithms are reviewed. In some applications, the similarity measure is obtained by comparing a predefined profile to the profile observed for each gene. This dissertation introduces such a data set from Chu *et al.* (1998) and reviews the prior analyses performed on these data. A new approach based on a smoothing mechanism is proposed for generating and comparing the profiles. Smoothing techniques applied prior to filtering along with several filtering techniques are discussed. Chu's (1998) data is reanalyzed using the new method. Finally, the smoothed and unsmoothed clustering results are compared graphically using the profile graphs.

2.1.1 Microarray Data Description

The amount and scope of microarray data is ever increasing. Microarray experiments are becoming cheaper and easier to perform with the help of popular designs such as the dye-swap, reference, and loop designs (Kerr and Churchill, 2001). As discussed in Chapter 1, the two major platforms on which microarray experiments are performed are the Affymetrix and complementary DNA (cDNA). The focus of this chapter is on the analysis of cDNA arrays.

In the case of cDNA microarrays, mRNA is extracted from the cells and hybridized to form the cDNA samples. These samples are usually labeled with a red (Cy3) or a green (Cy5) dye. Through a highly automated process, these samples are placed, or spotted, onto the microarrays. The identity of the spots is retained by keeping track of their position on the array. Standard identifying characteristics include the sample (*e.g.* the cell type or variety) and the gene from which the sample came. Each sample is typically spotted multiple times on an array in order to help to control technical variability. The microarrays are "read" by shining a laser through a particular spot on the array and recording a fluorescence value. The fluorescence value is indicative of the degree of gene expression activity for that sample.

Image analysis is the process of identifying signal and background from a scanned image. A common measure of the signal is foreground – background. The use of this background corrected value helps to separate the signal from the background. Normalization removes artifacts of non-biological origin such as print-tip effects, etc. (Quackenbush, 2001; Quackenbush, 2002). Typical analyses performed on the background corrected and normalized fluorescence values include cluster and classification analyses which may attempt to find genes which are expressed in specific biological processes. For more information on microarray technology, see Section 1.1.

2.1.2 The Use of Profiles in Microarray Data Analysis

In some experiments, researchers have prior knowledge about gene function relating to some process of interest. For example, suppose that several genes are known to influence a given biological process. Further suppose that these genes may be subdivided into groups according to which areas of biological activity they influence. Assume that the researcher is interested in using a microarray experiment to find additional genes that may be involved in the process. Standard cluster or classification analyses considering these genes could be performed in order to help group genes having similar expression values (Everitt *et al.*, 2001). Ideally, the resultant groups would contain genes involved in similar biological processes. However, by including known information regarding gene function, one can perhaps improve the interpretability of the gene clusters. In the literature, some researchers (Chu *et al.*, 1998; Kerr and Churchill, 2001) performed cluster analyses using profiles generated by known genes. In the case of Kerr and Churchill, the profile generating genes are placed into seven groups based on seven time points in the yeast sporulation process. Once the genes are grouped, average gene expression profiles are derived for each group. Other genes of unknown function from the microarray experiment are assigned to one of these profiles based on clustering using some distance or correlation metric which relates the gene of unknown function to the profile. The metric (1 - the Pearson correlation measure) is often used as the dissimilarity measure in performing profile based clustering of genes.

2.1.3 Gene Filtering

Many of the genes studied in a microarray experiment may not match well with any of the profiles. Including too many irrelevant genes can make the cluster results more noisy and thus reduces the chance of finding the small subsets of genes that may be involved in regulating specific processes. In order to improve the signal to noise ratio, gene filtering methods are commonly employed prior to running analyses. Many of these techniques are based on minimization of variance criteria. More advanced techniques include modeling approaches such as those discussed in Kerr and Churchill (2001), Rocke and Durbin (2001), and Wolfinger *et al.* (2001). One must be very careful not to choose a filtering technique that is so broad that it throws away genes of interest.

2.2 Similarity Measures

In order to identify clusters of observations that may be present in data, it is critical to have some measure of how "close" the observations are to each other. This "closeness" is quantified using a distance measure. A metric, δ_{ij}, fulfills the following triangle inequality:

$$\delta_{ij} + \delta_{im} \geq \delta_{jm} \tag{2.1}$$

for the pairs of individuals (i,j), (i,m), and (j,m) (Everitt *et al.*, 2001) where $\delta_{ij} = 0$ iff $i = j$ and $\delta_{ij} = \delta_{ji}$. The δ 's are always non-negative. A distance matrix may be defined where Equation 2.1 holds for all triplets (i,j,k) and $\delta_{ii} = 0 \ \forall i$. It can be seen from Equation 2.1 that the distance between individuals *i* and *j* is the same as that between *j* and *i* and that if two points *i* and *j* are close together then *k* must have a similar proximity to both of them. The Euclidean distance is one such metric.

Another measure known as the similarity measure is defined so that any two individuals are "close" when their similarity measure is large (or their dissimilarity measure is small). A similarity measure may be defined by reversing the inequality in Equation 2.1 and keeping the other conditions the same. Choosing a similarity measure is subjective and could vary according to the experiment (Johnson and Wichern, 1998). Important considerations when picking a measure include the nature of the variables (discrete, continuous, binary), scales of measurement (nominal, ordinal, interval, ratio), and subject matter knowledge.

2.2.1 Similarity Measures for Binary Data

A large number of similarity measures have been proposed for binary data. Table 2.1 illustrates the general form of this cross-classification.

Table 2.1: Counts of Binary Outcomes for Two Individuals

		Individual i		
	Outcome	1	0	Totals
Individual j	1	a	b	a + b
	0	c	d	c + d
	Totals	a + c	b + d	p = a + b + c + d

Table 2.2 gives a sample of the most commonly used similarity measures for binary data. A more extensive list can be found in Gower and Legendre (1986). Individual similarity measures are used to build an $n(n+1)/2$ dimensional similarity (or distance) matrix, where n is the number of observations that need to be clustered. The difference between the various similarity measures in Table 2.2 centers around how the 1-1 and the 0-0 matches are treated. For example, when comparing two genes in microarray data, if a 1 represents a gene expression value above the background level, the researcher is not as interested in the 0-0 matches as he/she would be in the 1-1 matches. This is because the 1-1 matches indicate that both genes are expressed above the background level simultaneously and thus may be involved

in similar biological functions. The investigator must decide on the relative importance of the 0-0 and the 1-1 matches based on the specific application.

Table 2.2: Similarity Measures for Binary Data

Measure	Rationale	Formula
Matching Coefficient	Equal weights for 1-1 matches and 0-0 matches.	$s_{ij} = \frac{a+d}{p}$
Jaccard Coefficient (Jaccard, 1908)	No 0-0 matches in the numerator or the denominator. (The 0-0 matches are treated as irrelevant.)	$s_{ij} = \frac{a}{a+b+c}$
Rogers and Tanimoto (1960)	Double weight for unmatched pairs.	$s_{ij} = \frac{a+d}{a+2(b+c)+d}$
Sokal and Sneath (1963)	No 0-0 matches in the numerator or the denominator. Double weight for unmatched pairs.	$s_{ij} = \frac{a}{a+2(b+c)}$
Gower and Legendre (1986)	Half weight for unmatched pairs.	$s_{ij} = \frac{a+d}{a+\frac{1}{2}(b+c)+d}$
Gower and Legendre (1986)	No 0-0 matches in the numerator or the denominator. Half weight for unmatched pairs.	$s_{ij} = \frac{a}{a+\frac{1}{2}(b+c)}$

2.2.2 Distance Measures for Continuous Data

The "difference" between two continuous observations is represented by two classes of measures: distance measures and correlation based measures. Distance measures, represented by d_{ij}, allow the interpretation of dissimilarities as functions of physical distances. Correlation measures, represented by δ_{ij}, are not straightforward functions of physical distance but are a type of similarity measure. Due to their simplicity in interpretation, distance measures are often preferred to similarity measures for continuous data. Table 2.3 shows several distance and correlation measures for continuous data (Everitt *et al*., 2001).

Table 2.3: Distance and Correlation Measures for Continuous Data

Measure	Formula
Euclidean Distance	$d_{ij}=\sqrt{\left(x_i-x_j\right)^T\left(x_i-x_j\right)}=\sqrt{\sum_{k=1}^{p}\left(x_{ik}-x_{jk}\right)^2}$
City Block Distance	$d_{ij}=\sum_{k=1}^{p}\left\|x_{ik}-x_{jk}\right\|$
Minkowski Distance	$d_{ij}=\left(\sum_{k=1}^{p}\left\|x_{ik}-x_{jk}\right\|^r\right)^{\frac{1}{r}}$, $r\geq 1$
Pearson Correlation	$\delta_{ij}=\left(1-\phi_{ij}\right)/2$ where $\phi_{ij}=\dfrac{\sum_{k=1}^{p}\left(x_{ik}-\bar{x}_{i.}\right)\left(x_{jk}-\bar{x}_{j.}\right)}{\sqrt{\sum_{k=1}^{p}\left(x_{ik}-\bar{x}_{i.}\right)^2\sum_{k=1}^{p}\left(x_{jk}-\bar{x}_{j.}\right)^2}}$ and $\bar{\mathrm{x}}_{i.}=\frac{1}{p}\sum_{k=1}^{p}x_{ik}$ $\quad \bar{x}_{j.}=\frac{1}{p}\sum_{k=1}^{p}x_{jk}$

Legend: p = number of variables
x = vector of continuous responses for a given observation

2.3 Hierarchical Clustering Methods

There are two basic types of hierarchical clustering methods. The first type is the agglomerative hierarchical method. These methods start with as many clusters as objects and merge objects into groups according to their similarities (or dissimilarities). In the first iteration, the most similar objects are grouped together and merged. In the final iteration, all of the objects are contained in a single large cluster. Sometimes the objects in the final cluster are quite dissimilar.

The second type of hierarchical clustering method is the divisive hierarchical method. These methods start with all of the objects contained in one large cluster. At each iteration, the groups are subdivided or kept in the same cluster based on how close the individuals within clusters are in terms of their similarity measures. Eventually, there are as many clusters as individuals.

2.3.1 Agglomerative Hierarchical Clustering Methods

Agglomerative clustering methods are among the most widely used clustering techniques. These methods partition the data such that the first partition consists of *n* clusters, each containing a separate observation, and the last partition consists of a single cluster containing all *n* observations. At each iteration, the methods fuse the observations (or the groups of observations) which are the most similar. The methods differ in how the similarities (or the distances) are calculated between the clusters.

The application of the methods is quite similar. Johnson and Wichern (1998) give the general steps in agglomerative clustering algorithms for grouping *N* objects as:

1. Start with N clusters, each containing a single entity and an NxN symmetric matrix of distances (or similarities) **D** = $\left|d_{ik}\right|$.
2. Search the distance matrix for the nearest (*i.e.* most similar) pair of clusters. Let the distance between "most similar" clusters U and V be d_{uv}.
3. Merge clusters U and V. Label the newly formed cluster (UV). Update the entries in the distance matrix by (a) deleting the rows and columns corresponding to clusters U and V and (b) adding a row and column giving the distances between cluster (UV) and the remaining clusters.
4. Repeat steps 2 and 3 a total of N-1 times. Recall that all of the objects will be in a single cluster at the termination of the algorithm. Record the identity of clusters that are merged and the levels at which the merges take place.

There are various ways of calculating distances between an object and a cluster or between two clusters for agglomerative clustering methods. These clustering techniques are discussed below and include single linkage, complete linkage, average linkage, and Ward's method.

Single Linkage

The inputs to a single linkage algorithm are distances or similarities between pairs of objects. This clustering method is also known as the nearest neighbor method because clusters are formed from individual objects by merging the nearest neighbors, or those objects with the smallest distance or largest similarity between each other. Single linkage was first proposed by Sneath (1957).

The single linkage algorithm is illustrated with an example. The data for this example is from a hypothetical microarray experiment. The goal of this experiment is to group genes having similar expression patterns across three tumor cell lines. The fluorescence values are shown in Table 2.4.

Table 2.4: Data from an Example Microarray Experiment

Gene Number	Cell Line 1	Cell Line 2	Cell Line 3
1	0.095516	0.077364	-0.250570
2	-0.075721	0.499687	0.045323
3	0.195900	0.281033	0.012837
4	-0.267606	-0.301030	0.993877
5	0.338457	0.318063	0.526339

Figure 2.1 gives a distance matrix containing Pearson correlation dissimilarity measures, which are described in Section 2.2 as (1 – the usual pair wise Pearson correlation coefficient).

$$\mathbf{D} = \{d_{ik}\} = \begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
1 & 0.000 & & & & \\
2 & 0.727 & 0.000 & & & \\
3 & 0.065 & 0.404 & 0.000 & & \\
4 & 1.998 & 1.339 & 1.957 & 0.000 & \\
5 & 1.991 & 1.400 & 1.974 & \boxed{0.002} & 0.000
\end{array}$$

Figure 2.1: Distance Matrix for Single Linkage Microarray Example, First Iteration

Notice that the element with the smallest off-diagonal value has a box drawn around it. Since $\min\limits_{i,k}\left(d_{ik}\right) = d_{54} = 0.002$, observations 4 and 5 are merged together to form cluster (4 5). The distances between this new cluster and all existing clusters must be calculated. These "nearest neighbor" distances are:

$$d_{(45)1} = \min\left[d_{41}, d_{51}\right] = \min\left[1.998, 1.991\right] = 1.991$$
$$d_{(45)2} = \min\left[d_{42}, d_{52}\right] = \min\left[1.339, 1.400\right] = 1.339$$
$$d_{(45)3} = \min\left[d_{43}, d_{53}\right] = \min\left[1.957, 1.974\right] = 1.957$$

Plugging in these new values yields the distance matrix shown in Figure 2.2.

$$\mathbf{D} = \{d_{ik}\} = \begin{array}{c|cccc|}
 & (4\ 5) & 1 & 2 & 3 \\
(4\ 5) & 0.000 & & & \\
1 & 1.991 & 0.000 & & \\
2 & 1.339 & 0.727 & 0.000 & \\
3 & 1.954 & \boxed{0.065} & 0.404 & 0.000
\end{array}$$

Figure 2.2: Distance Matrix for Single Linkage Microarray Example, Second Iteration

Once again, the element with the smallest off-diagonal value has a box drawn around it. Observations 1 and 3 are merged together to form cluster (1 3). This process continues until all of the observations are contained in one cluster of size *N*. The output of a cluster analysis is typically displayed in a graph called a dendrogram. Figure 2.3 shows the output of the single linkage clustering microarray example.

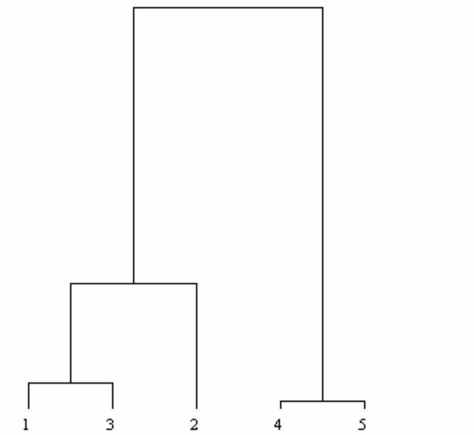


Figure 2.3: Results of Single Linkage Cluster Analysis for the Microarray Example

The distance between the nodes in the dendrogram represents how uniform the cluster members are. Clusters having small distances between nodes are more uniform, or “tighter”. Notice that at the bottom of the tree each individual is contained within its own cluster and at the top of the tree, or “root cluster”, all of the objects are placed into the same cluster. The analyst must decide at which level of clustering to interpret the data. Further investigation of the relationships discovered using cluster analysis is often warranted. One problem with the single linkage method is that it tends to produce unbalanced and straggly clusters, especially in large data sets, and does not take cluster structure into account (Everitt *et al.*, 2001).

Complete Linkage

Complete linkage clustering is similar to single linkage clustering, except that at each stage the clusters are formed by choosing the most distant observations from each other. This method ensures that all of the items in a cluster are within some maximum distance (or minimum similarity) of each other. This method was first proposed by Sorensen (1948). Since this method proceeds identically to the single linkage algorithm with the exception of updating the distance matrix based on the maximum distances, the algorithm is not discussed in detail. Complete linkage clustering using the distance matrix first presented in Figure 2.1 yields the dendrogram shown in Figure 2.4. Observe by comparing Figures 2.3 and 2.4 that, for this example, application of the single and complete linkage methods results in identical clusters. This is not always the case.

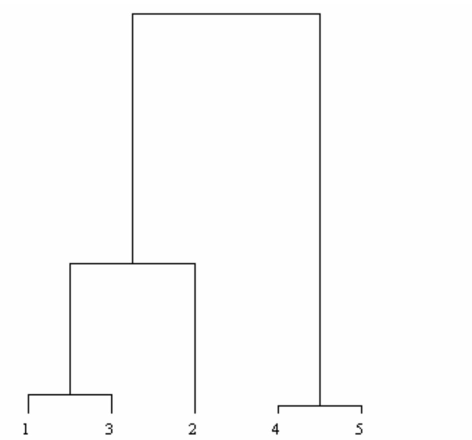


Figure 2.4: Results of Complete Linkage Cluster Analysis for the Microarray Example

One must be careful when selecting the type of cluster analysis to use and perhaps perform the analysis using multiple clustering algorithms for the sake of comparison. A problem with the complete linkage method is that it tends to find compact clusters with equal diameters and does not take into account cluster structure (Everitt *et al.*, 2001).

Average Linkage

The average linkage clustering technique proceeds very similarly to the single and complete linkage clustering techniques. The objects are clustered according to the smallest distances (or largest similarities) calculated using Equation 2.2. This method was first proposed by Sokal and Michener (1958).

The difference in the average linkage method lies in how to calculate the distances between a newly formed cluster and all of the original clusters. Suppose that a new cluster, (UV), has been formed based on the minimum distance criteria. In the average linkage algorithm, the distance between cluster (UV) and a cluster W is found by the formula:

$$d_{(UV)W} = \frac{\sum_{i}\sum_{k} d_{ik}}{N_{(UV)} N_{W}} \tag{2.2}$$

where d_{ik} is the distance between object *i* in the cluster (UV) and object *k* in cluster W, and $N_{(UV)}$ and N_W are the numbers of items in clusters (UV) and W, respectively.

The average linkage clustering method is applied to the microarray example presented in Table 2.4. This results in the dendrogram presented in Figure 2.5. Notice that these results are identical to the previous results. This is not always the case. A limitation of the average linkage clustering technique is that it tends to join clusters with small variances (Everitt *et al.*, 2001). However, it represents a good compromise between the single and complete linkage clustering methods. Additionally, average linkage takes cluster structure into account and is relatively robust.

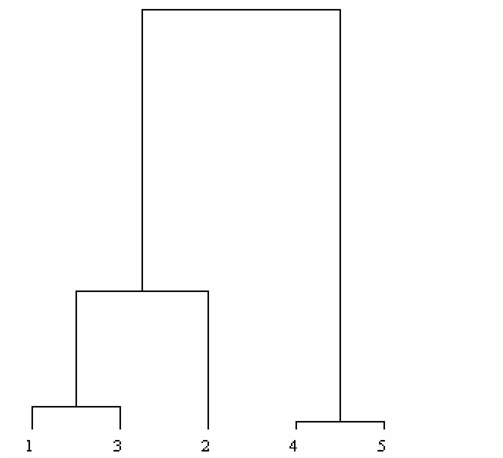


Figure 2.5: Results of Average Linkage Cluster Analysis for the Microarray Example

Ward's Method

Ward (1963) first proposed this popular method which merges clusters based on the size of an error sum of squares criterion. This error sum of squares criterion is based on the amount of information lost when two groups are joined. Loss of information increases the error sum of squares (ESS). This algorithm, as described in Johnson and Wichern (1998), is given below.

For a given cluster *k*, let ESS_k be the sum of squared deviations of every item in the cluster from the cluster mean. If there are currently *C* clusters, ESS is defined as $ESS = \sum_{k=1}^{C} ESS_k$. At each step in the analysis, the union of every possible pair of clusters is considered, and the two clusters whose combination results in the smallest increase in ESS are joined. Initially, each cluster consists of a single item, thus ESS = 0. When the algorithm terminates and all of the clusters are combined into a single group of N items,

$ESS = \sum_{j=1}^{N} \left(x_j - \overline{x}\right)' \left(x_j - \overline{x}\right)$, where x_j is the data from the j^{th} item and $\overline{x}$ is the mean of all of the items. Applying Ward's method to the microarray data presented in Table 2.4 results in a clustering identical to the average linkage method results. This is not always the case.

Ward's method is based on the notion that the clusters of observations are expected to be elliptically shaped. It assumes that points can be represented in Euclidean space, tends to find spherically shaped clusters of the same size, and is sensitive to outliers (Everitt *et al.*, 2001). Monte Carlo simulation studies have demonstrated that Ward's method is as good or better than other hierarchical techniques (Kuiper and Fisher, 1975; Blashfield, 1976; Mojena, 1977; Milligan, 1980; Breckenridge, 1989), particularly under high noise conditions.

2.3.2 Divisive Hierarchical Clustering Methods

Divisive hierarchical clustering methods work in the opposite way of agglomerative hierarchical clustering methods in the sense that they start with one cluster containing all of the objects and successively split this cluster into smaller clusters. According to Everitt *et al.* (2001), agglomerative methods are much more commonly used than divisive methods. However, Kaufman and Rousseeuw (1990) point out that divisive clustering methods have the advantage of being able to see the structure in the data as it is discovered. Divisive clustering methods are not the focus of this chapter and more information about them may be found in Everitt *et al.* (2001).

2.3.3 Final Comments on Hierarchical Clustering Methods

There are several items that one must consider when using hierarchical clustering methods. The first of these is the choice of the clustering method. Choices also must be made regarding which distance or similarity measure to use. One source of frustration in applying clustering methods is that different clustering techniques can lead to quite different results. Thus, interpretation of the cluster results is often difficult. Generally, an investigator chooses a level of clustering in the hierarchy that seems to make sense in the context of the problem. Like all statistical techniques, there are choices and tradeoffs to be considered when selecting a method to use.

In some applications, it may make sense to allow individuals to be contained in more than one cluster. For example, if noisy data is present, there may not be enough information to obtain an adequate separation of clusters. In cases like this, the best the researcher can do may be to say that an individual belongs to cluster A and/or cluster B. Forcing these individuals into one cluster or the other could mask an existing relationship. Another possible problem arises from the fact that once divisions or fusions are made, they cannot be reversed. Finally, hierarchical clustering methods always produce a clustering result regardless if there is any actual pattern present in the data.

2.4 Nonhierarchical Clustering Methods

Nonhierarchical clustering methods do not force a hierarchical clustering structure and thus allow for data lacking any inherent hierarchy. Many of these methods require the user to specify in advance the number of clusters desired in the output. One way to deal with this limitation is to do several analyses with different numbers of clusters and see which clustering scheme results in the minimal variance of its members (Everitt *et al.*, 2001).

2.4.1 Optimization Based Clustering Methods

Once a clustering criterion has been chosen, it needs to be repeatedly applied to assign the objects to clusters. The best way to do this is to calculate the value of the clustering criterion for every possible partition and to choose the partition having the best value. However, as discussed in Section 1.2, this is not computationally feasible due to the huge number of possible clusters in even a moderately sized problem. Liu (1968) gives the following formula for calculating the number of possible partitions of *n* objects into *g* groups:

$$N\left(n,g\right)=\frac{1}{g\,!}\sum_{m=1}^{g}\left(-1\right)^{g-m}\binom{g}{m}m^{n}\ . \tag{2.3}$$

For example, N(10,5) = 42,525. Calculating all of the possible partitions for a problem of this size seems feasible. However, consider a realistic problem of clustering 60 cancer cell lines into 4 clusters. N(60,4) = $5.54x10^{34}$. Calculating all of these possibilities is not possible on today’s computers in any reasonable length of time.

Algorithms which search through the possible cluster assignments and keep only the ones that improve the value of the clustering criterion are called hill-climbing algorithms. These algorithms have some possible limitations, such as finding local maximums or minimums or not converging quickly enough. However, hill-climbing algorithms are widely used in optimization. These techniques share common steps described by (Everitt *et al.*, 2001) as:

1. Find some initial partition of the *n* objects into *g* groups.
2. Calculate the change in the clustering criterion produced by moving each object from its own group to another group.
3. Make the change which leads to the greatest improvement in the value of the clustering criterion.
4. Repeat the previous two steps until no move of a single object causes the clustering criterion to change.

One of the most commonly used optimization techniques for clustering is the k-means method, which is discussed below.

K-means Clustering Algorithm

MacQueen (1967) developed an algorithm which assigns each item to the cluster having the nearest mean. This algorithm is called the k-means algorithm, and remains one of the most popular clustering algorithms. Johnson and Wichern (1998) described the steps of the k-means algorithm as:

1. Partition the items into *C* initial clusters.
2. Proceed through the list of items, assigning an item to the cluster whose centroid is nearest. A distance measure must be chosen. Recalculate the centroid for the cluster receiving the new item and for the cluster losing the item.
3. Repeat step 2 until no more reassignments take place.

This algorithm is illustrated with an example using the microarray data from Table 2.4. Suppose that the investigator wishes to partition the data into 3 clusters. For illustration purposes, the Euclidean distance measure is used. The initial partition is arbitrarily chosen to be (1,3), (2,4), and (5). These partitions along with the associated mean vectors are shown in Table 2.5.

Table 2.5: Initial Partitions for the K-means Algorithm Microarray Example

Cluster	Individual	Mean Vector
1	1, 3	(0.146, 0.179, -0.119)
2	2, 4	(-0.172, 0.099, 0.520)
3	5	(0.338, 0.318, 0.526)

Now the distance from each individual to each cluster's mean vector is calculated. If the distance is smaller than that of the individual to the current cluster, the individual is reassigned to the closer cluster. This process is illustrated below.

First, the distance between the first individual and the cluster means is calculated.

$$d^2\left[(1),(1,3)\right]=\left(0.095-0.146\right)^2+\left(0.077-0.179\right)^2+\left(-0.251+0.119\right)^2=0.030$$
$$d^2\left[(1),(2,4)\right]=0.666$$
$$d^2\left[(1),(5)\right]=0.721$$

The distance of 0.030 between cluster (1) and cluster (1,3) is the smallest of the three distances, so (1) is already contained in the correct cluster at this stage. Now the distance between (2) and the cluster means is calculated.

$$d^2\left[(2),(1,3)\right]=0.179$$
$$d^2\left[(2),(2,4)\right]=0.396$$
$$d^2\left[(2),(5)\right]=0.436$$

Individual (2) is closest to cluster (1,3), so a new cluster (1,2,3) is formed. Table 2.6 is recalculated for these new clusters.

Table 2.6: Partitions for the K-means Algorithm Microarray Example, Second Iteration

Cluster	Individual	Mean Vector
1	1, 2, 3	(0.072, 0.286, -0.064)
2	4	(-0.268, -0.301, 0.994)
3	5	(0.338, 0.318, 0.526)

The process iterates, with the distance between each individual and the clusters being calculated and the cluster assignment based on minimizing this distance. At termination, the final cluster assignments for the k-means method were (1,2,3), (4), and (5). The dendrogram is given in Figure 2.6.

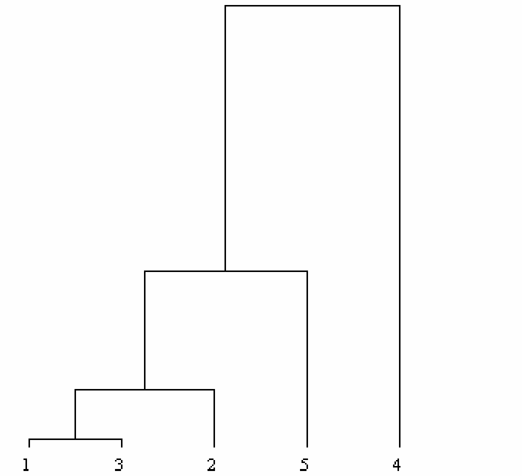


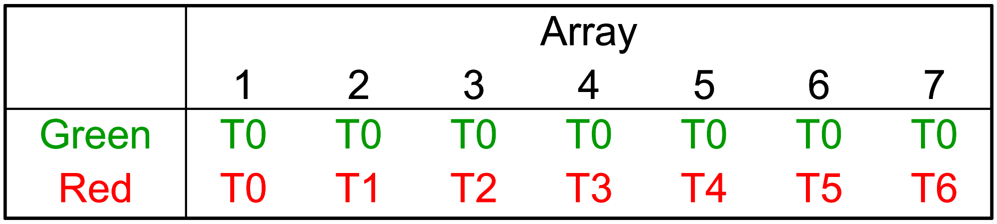
Figure 2.6: Results of K-Means Cluster Analysis for the Microarray Example

2.4.2 Other Techniques

There are a variety of nonhierarchical techniques available for cluster analysis that do not fall into a convenient grouping scheme. Many of these techniques are specialized and have computer science roots in areas such as pattern recognition and artificial intelligence. Several of these techniques, such as self organizing maps and genetic algorithms, are discussed briefly in Chapter 6.

2.5 Introduction to Chu *et al.* (1998) Microarray Data

The data for this analysis are from the experiment reported in Chu *et al.* (1998). In this experiment, spotted cDNA microarrays containing 97% of the known genes of Saccharomyces cerevisiae (yeast) were used to study gene expression during meiosis and spore formation. Yeast cells were transferred to a nitrogen-deficient medium to induce sporulation and mRNA samples were taken at seven time points: 0, 30 minutes and 2, 5, 7, 9, and 12 hours. The ‘‘varieties’’ in this experiment are the time points. For each time point, the scientists prepared a ‘‘red’’ labeled cDNA pool. In addition, they prepared a ‘‘green’’ labeled cDNA pool from the time-0 sample. Seven microarrays were used in the study, one for each of the seven time points. Each array was probed with the green-labeled sample mixed with one of the seven red-labeled samples. Figure 2.7 shows the design of the experiment, which is known as an augmented reference design.



	Array						
	1	2	3	4	5	6	7
Green	T0	T0	T0	T0	T0	T0	T0
Red	T0	T1	T2	T3	T4	T5	T6

Figure 2.7: Design of the Augmented Reference Microarray Experiment

Notice in Figure 2.7 that variety (or time point) 0 is repeated 7 times using the green dye and once using the red dye. In effect, time 0 serves as a reference for all of the samples. All other varieties appear one time each using the red dye. The design is augmented because of the variety 0 red-dye block. If this block were another variety, *e.g.* variety 7, then the design would be the standard reference design. Note that each “block” represents all 6,118 genes and that the design is balanced with respect to arrays and dyes.

The data set contains four fluorescence measurements for each spot: green signal, green background, red signal, and red background. As their estimate of the relative expression of a gene at time *k* compared with time 0, Chu *et al.* (1998) used the background-corrected ratio (red signal - red background)/(green signal - green background) from the array containing red-labeled cDNA from time *k* and green-labeled cDNA from time 0.

Prior Analyses

Chu *et al.* (1998) performed the original experiment and rudimentary statistical analyses. They constructed seven profiles using genes previously found to be involved in sporulation. These genes are shown in Table 2.7 and range over the metabolic, early I, early II, early-middle, middle, middle-late, and late phases of sporulation.

Table 2.7: Genes Used to Create Average Temporal Profiles

Metabolic	Early I	Early II	Early-Mid	Middle	Mid-Late	Late
ACS1 PYC1 SIP4 CAT2 YOR100C CAR1	ZIP1 YDR374C DMC1 HOP1 IME2	KGD2 AGA2 YPT32 MRD1 SPO16 NAB4 YPR192W	YBL078C QRI1 PDS1 APC4 KNR4 STU2 YNL013C EXO1	YSW1 SPR28 SPS2 YLR227C ORC3 YLL005C YLL012W	CDC27 DIT2 DIT1	SPS100 YKL050C YMR322C YOR391C

The Pearson correlations between the individual genes and each of the seven profiles were calculated. The unknown genes were assigned to the profile for which their Pearson correlation was the highest. Once this clustering was done, the correlations were ranked for each of the groups. The bulk of the paper is a biological commentary on the significance of the top ranked genes in a given group.

Kerr and Churchill (2001) reanalyzed Chu's data using a model based approach. They used the same seven profiles and fitted an ANOVA model containing terms for array, dye, array x dye, gene, array x gene, and variety x gene effects. Gene filtering was done using the estimates of the variety x gene interaction affects from the ANOVA. There were 130 genes kept. The genes used to generate the profiles were not included in these 130 genes. Once the filtering method was applied, the genes were assigned to the group for which their Pearson correlation was a maximum. Kerr and Churchill performed bootstrapping to determine how stable the cluster assignments were. Figure 2.8 gives the 95% stable cluster result profiles that they reported.

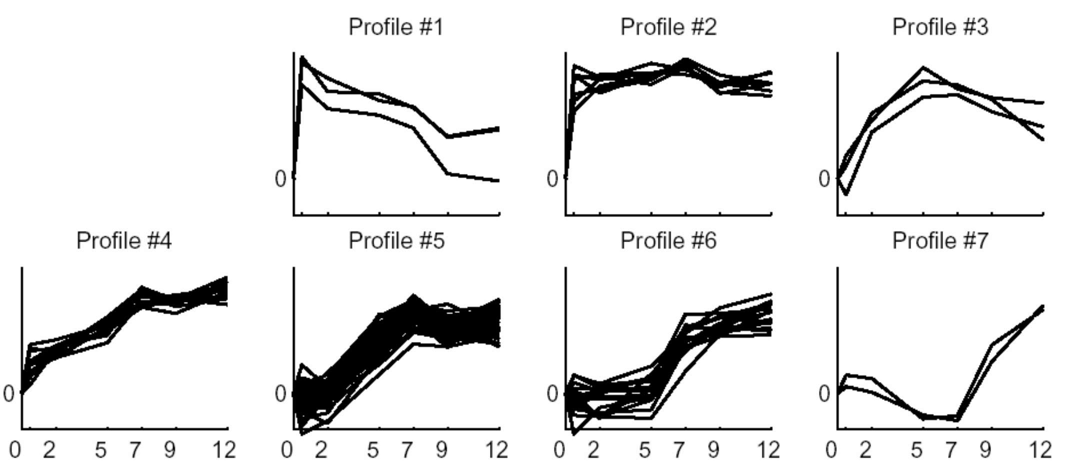


Figure 2.8: Seven 95% Stable Cluster Profiles from Kerr and Churchill (2001)

Table 2.8 shows the number of genes (out of 130) assigned to each profile.

Table 2.8: Number of Genes Assigned to Each Profile

Profile Number	1	2	3	4	5	6	7
Number of Genes	3	7	3	12	86	17	2

Neither Figure 2.8 nor Table 2.8 offer any information about the specific genes that group together. Such information is necessary to make biological hypotheses about why the genes group together.

2.6 Gene Smoothing

2.6.1 Smoothing Techniques

Smoothing techniques are data transformations made to lessen the impact of individual observations on the overall pattern or “shape” of the data. Smoothing can help to remove “spikes” from the data in order to focus on the signal and can be useful for comparing noisy data sets. A wide variety of smoothing techniques are available. The simplest smoothers are based on transformations such as $\sqrt{\ }$ or $X^{\frac{1}{N}}$, where N is an integer. Logarithmic transformations are also used as data smoothers. More complex smoothing methods include the local regression (LOESS) and penalized least squares (referred to as TPSPLINE in SAS) approaches, as well as more complex kernel based methods.

The local regression, or LOESS, smoothing method has become quite popular due to its ease of implementation and ability to work well in a wide variety of situations. LOESS makes no assumptions about the parametric form of the regression surface. The form of the LOESS regression model is: $y_i = g\left(x_i\right) + \varepsilon_i$, where $i = 1,\ldots,n$, y_i is the i^{th} response, x_i is the i^{th} vector of p predictors, g is the regression function, and ε_i is a random error. The idea behind local regression is that the regression function $g\left(x\right)$ at a specific x can be locally approximated by the value of a function in some specified parametric class (SAS Institute Inc., 1999). The local approximation is found by fitting a regression model to the data points within a given neighborhood of the point x, $\left[x - \omega, x + \omega\right]$. Weighted least squares is used to fit linear or quadratic functions of the predictors. The radius is chosen so that the neighborhood contains a specified percentage of the data points. A SAS option in the LOESS procedure which automatically selects the "best" radius was applied (SAS Institute Inc., 1999). For additional information on LOESS, see Cleveland and Grosse (1991). The TPSPLINE approach requires more data points than LOESS and cannot be easily applied to Chu's data since there are only seven observations for each gene. TPSPLINE is not discussed in this dissertation.

2.6.2 Smoothing Applied to Chu *et al.* (1998) Microarray Data

The data from Chu *et al.* (1998) is used to illustrate the methods discussed. In order to more accurately match gene expression levels across the seven time points with a given profile, a LOESS smoothing model is fit for each gene. For each temporal pattern, the model has seven independent variables coming from the expressions for the set of genes at each of the seven time points. The model predicts smoothed values for each time point and essentially replaces the original data with their smoothed counterparts. The local regression terms are allowed to be quadratic. New profiles are constructed using these smoothed values. The seven profiles are constructed from the genes listed in Table 2.7. The smoothed gene expression profiles are visually compared with the smoothed profiles. The independent variables are the $\log\left[\left(\text{red signal - red background}\right)/\ \left(\text{green signal - green background}\right)\right]$ corrected values used by Kerr and Churchill (2001).

Running the individual LOESS models for each of the 6,188 genes takes approximately 30 minutes on a 1.5 gigahertz Microsoft Windows XP machine and could be sped up by using compiled code. The SAS code for this operation is given in Appendices 2.1 and 2.2. Figure

2.9 shows the unsmoothed profiles for the seven sporulation phases. The profiles were generated from the average expression levels for the genes listed in Table 2.7.

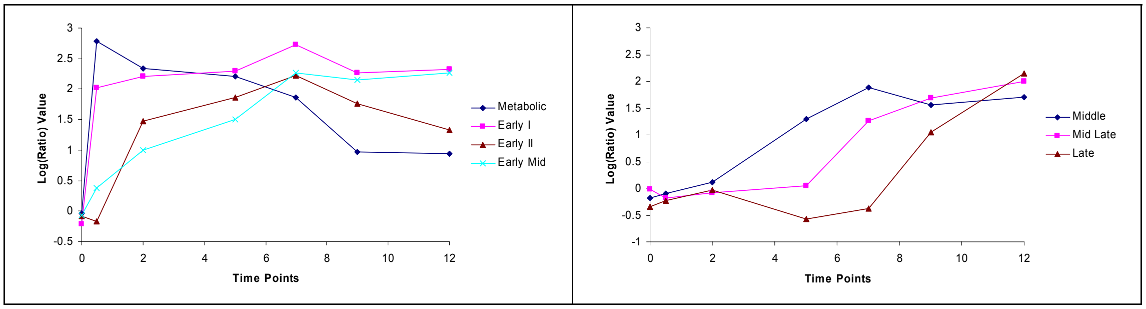


Figure 2.9: Unsmoothed Profiles

Figure 2.10 shows the same profiles after smoothing the genes using LOESS.

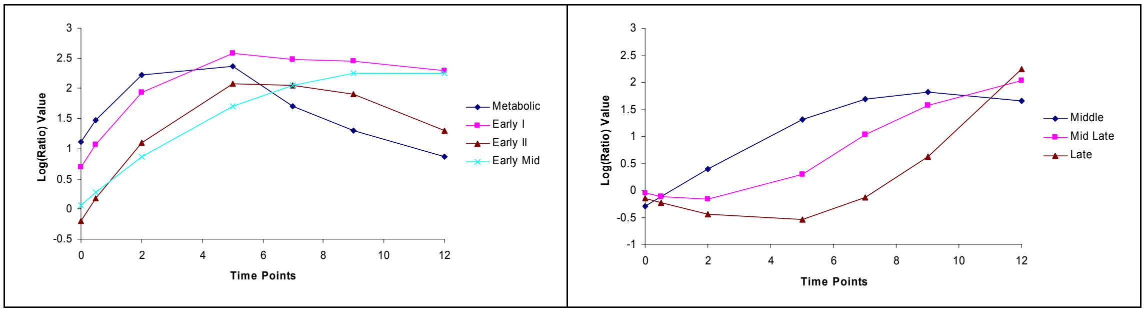


Figure 2.10: LOESS Smoothed Profiles

Notice that smoothing increases the separation between the profiles. For example, for the Metabolic and Early I phases at time points 0 and 30 minutes, the profiles were nearly coincident when unsmoothed (Figure 2.9) but are more distinct when smoothing is applied (Figure 2.10). Changes in gene expression levels happen gradually, and the smoothed profiles may more closely resemble what occurs in nature. Filtering relies on the uniqueness of the profiles, and the smoothed profiles may help to improve the cluster results by further distinguishing the profiles where possible.

2.7 Gene Filtering

2.7.1 Gene Filtering Techniques

Microarray technologies assess large numbers of genes. Some of these genes (such as genes for spike-in controls) may not be relevant to the researcher's questions. Filtering techniques allow non-informative observations to be removed from the data set prior to the analysis. One must be careful not to use a filtering method that is so stringent that it excludes informative observations. Along with thoughtful experimental design, a good filtering method can be instrumental in controlling noise in data.

A commonly used method for gene filtering is based on the variability of gene expression values for a given gene. Genes whose expression values do not change by more than a specified value across the samples are filtered out. The logic behind this type of filtering is that gene expression values for a gene active in a specific biological process should change at some point. Filtering based solely on variability works well in some cases. However, no consideration is given to the baseline levels of gene expression. Genes that are naturally lowly expressed will have small variances which could result in their being filtered out. These genes could be incorrectly removed if a small change in expression values is biologically significant. Specifying a single variance threshold for determining whether or not to keep a gene implies the assumption that all of the genes have similarly scaled variances. This may not always be true. For example, a gene with a large mean and variance for its expression level is of less interest than a gene with a small mean and the same large variance. One might consider using a filtering technique based on the coefficient of variation in this situation. The coefficient of variation gives a measure of the variability in relation to the magnitude of the estimate and is calculated by dividing the estimate by its standard error.

Chu *et al.* (1998) applied a simple filtering method to the yeast sporulation data. The genes were grouped according to which profile their expression levels were maximally correlated. The maximum Pearson correlations for each gene were ranked after the genes were assigned to groups. An arbitrary cutoff was chosen for the correlations, and genes having correlations below this threshold were excluded.

Chu's method of filtering has several potential problems. One possible source of bias is that genes having similar correlations with more than one profile are always assigned to the profile for which the correlation is the maximum. This process ignores the fact that clustering is not an exact science and a more accurate clustering could result from assigning the gene to a cluster for which its correlation is slightly weaker. Another issue is that the profiles are constructed by averaging expression levels of their component genes. This could mask some of the characteristics of the individual genes belonging to the profile. A single unknown gene is compared with a composite profile. Deviations from this profile may overly influence a correlation measure. Possible solutions to this problem include using different distance or similarity measures, different clustering algorithms, or transforming the data in some way prior to clustering.

Kerr and Churchill (2001) fitted an ANOVA model containing fixed effect terms for array, dye, array x dye, gene, array x gene, and variety x gene effects. Once this model was fitted, the difference in gene expression for gene *g* at time *k* compared with time 0 was estimated along with 99 percent bootstrap confidence intervals for these estimates. Gene filtering was performed using the criteria that this difference must be greater than zero for at least one time point *k* not zero and that the confidence interval for the difference does not contain zero. If these conditions were not met for a given gene, the gene was excluded from further analysis.

This analysis also compares a single gene to a composite profile. The model based approach forces only certain terms to be included in the analysis. Other effects involved could be left out. Due to the amount of data present, this model must be fit in stages (Kerr and Churchill, 2001). This approach works for a simple model, but becomes quite difficult for more complex models like the mixed effects model. Finally, bootstrapping is computationally intensive and may not be practical for all researchers.

2.7.2 Filtering Applied to Chu *et al.* (1998) Microarray Data

A novel approach to filtering microarray data based on the profiles presented in Section 2.6.2 is discussed in this section. The new method is similar to the method proposed by Chu *et al.* (1998) since it is also based on the Pearson correlations between the gene expression values for the individual genes and the seven profiles. Seven Pearson correlation coefficients are calculated for every gene (one for each profile). This results in a 6,118 x 7 (number of genes by number of profiles) matrix of correlations. Each of the seven columns is individually ranked from highest to lowest correlation. This is different from Chu's (1998) method since they found the maximum correlation for each of the seven profiles and based the filtering on a ranked list of these values. The SAS code for this filtering method is given in Appendix 2.3. A threshold value is chosen and genes that have Pearson correlations less than this value are filtered out. Genes with Pearson correlations above this value are kept. Table 2.9 illustrates the filtering process. The X's in the grid have a gene identifier associated with them and represent the Pearson correlations between a gene and the profile associated with the column number.

Table 2.9: Gene Filtering Using 7 Profiles

	Profile Number						
	1	2	3	4	5	6	7
Genes Kept	X	X	X	X	X	X	X
	X	X	X	X	X	X	X
	X	X	X	X	X	X	X
	X	X	X	X	X	X	X
Genes Filtered Out	X	X	X	X	X	X	X
	X	X	X	X	X	X	X
	X	X	X	X	X	X	X
	X	X	X	X	X	X	X
	X	X	X	X	X	X	X

Each profile has a set of genes which are kept since their correlations are above the threshold value. Genes having high correlations with more than one profile potentially appear more than once in the list of genes kept after the filtering process. Duplicates are removed from the list of genes.

For illustration purposes, filtering is performed on the Chu *et al.* (1998) microarray data. Filtering is done twice, once for the original unsmoothed data and once for the LOESS smoothed data. Each filtering application uses two different threshold values. The first threshold applied is stringent and keeps only the top 20 genes in each of the seven profiles. This means that there is a maximum of 20 x 7 = 140 genes kept (if they are all unique). The second threshold is less stringent and keeps the top 100 genes in each of the seven profiles. This implies a maximum of 100 x 7 = 700 genes kept. Note that one could include a different number of genes from each time point. One way to do this would be to base the gene selection on a correlation threshold. For example, one might only select the genes that had a correlation with a profile of greater than 0.8.

For the original unsmoothed data, filtering resulted in 140 genes kept for the cutoff value of 20. None of the genes kept were duplicated in the other profiles. Kerr and Churchill (2001) kept 130 genes after applying their filter. They did not list the individual genes selected. Applying the less stringent filter using a cutoff value of 100 resulted in 666 genes being kept. For this filter, 4.9 percent of the genes kept were duplicated in other profiles.

For the LOESS smoothed data, filtering resulted in 140 genes kept for the cutoff value of 20. None of the genes kept were duplicated in the other profiles. Applying the less stringent filter using a cutoff value of 100 resulted in 619 genes being kept. For this filter, 11.6 percent of the genes kept were duplicated in other profiles. The cluster analysis results obtained after filtering the data are presented in the next section.

2.8 Cluster Analysis of Chu *et al.* (1998) Microarray Data

Cluster analysis is performed using the average linkage and k-means techniques. The Pearson correlation similarity measure was used and seven clusters were requested. Figures 2.11 and 2.12 show the seven clusters generated by the average linkage and k-means clustering algorithms for filtering cutoff values of 20 and 100 (discussed in Section 2.7.2) for both the smoothed and the unsmoothed data. The seven clusters from the average linkage method were selected by "cutting" the dendrogram at the seven cluster level. Our results are not necessarily comparable to Kerr and Churchill's (2001) results (Figure 2.8), as we applied a different filtering method.

The first cluster analysis is performed on the unsmoothed data using a threshold value of 20. Table 2.10 gives the number of elements assigned to each cluster by the average linkage and k-means clustering algorithms. All of the cluster graphs are on the same scale.

Table 2.10: Cluster Sizes for the Average Linkage and K-means Techniques
(Filtering Threshold = 20, Unsmoothed)

Cluster Number	Average Linkage	K-means
1	20	36
2	40	19
3	20	23
4	20	23
5	10	10
6	10	19
7	20	10

Figure 2.11 contains plots of the seven clusters for each method. The x-axis represents the time point and the y-axis represents the gene expression value. Since cluster labeling is arbitrary, an attempt is made to place the most similar clusters (by visual inspection) from each method side by side.

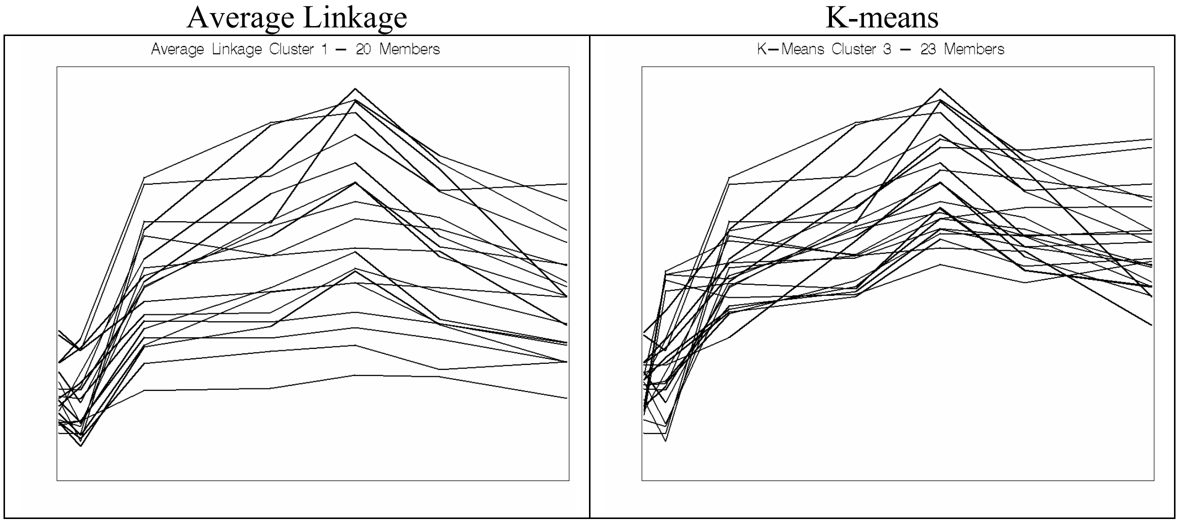


Figure 2.11: Average Linkage and K-Means Clusters
(Filtering Threshold = 20, Unsmoothed)

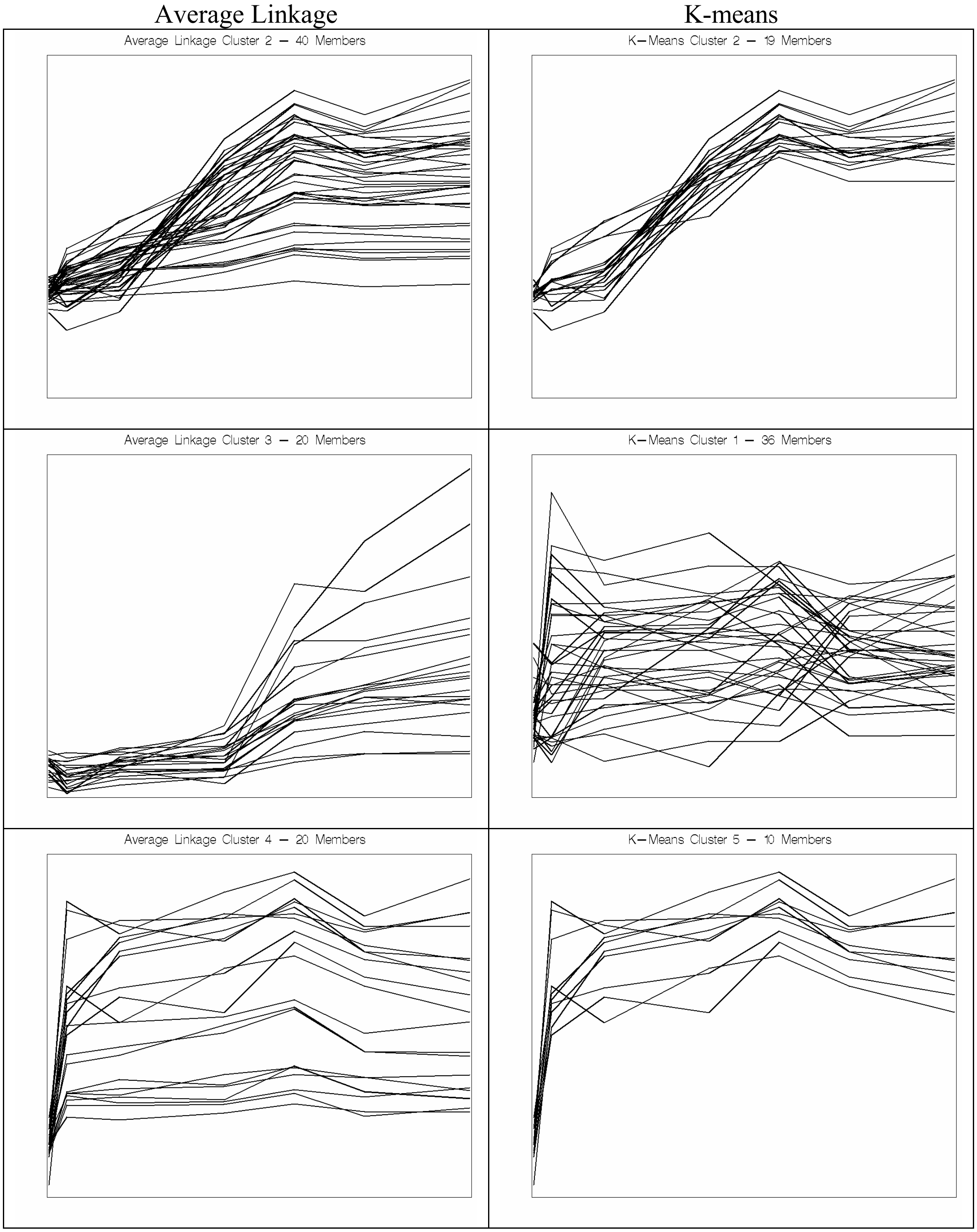


Figure 2.11: Average Linkage and K-Means Clusters
(Filtering Threshold = 20, Unsmoothed)
(continued)

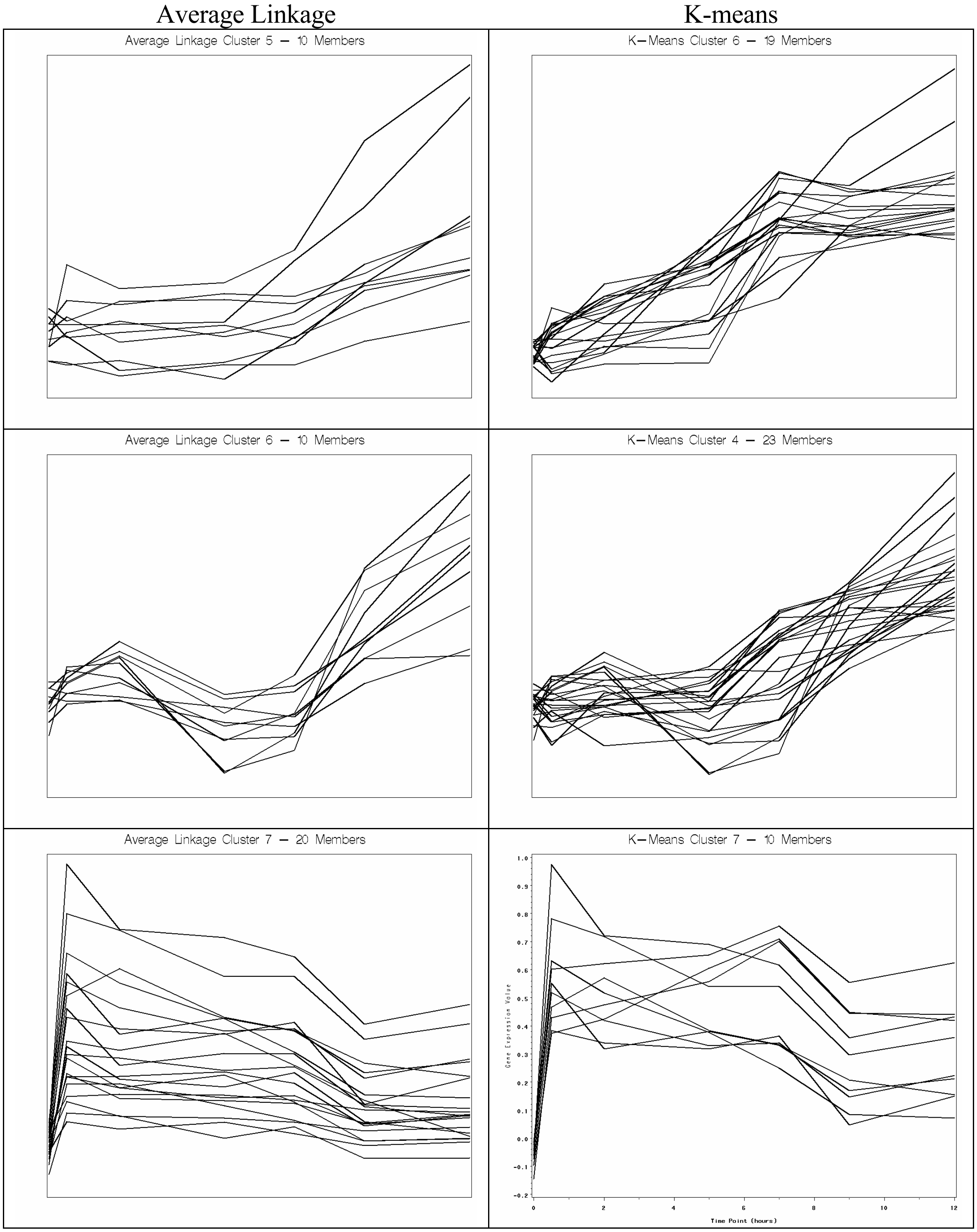


Figure 2.11: Average Linkage and K-Means Clusters (Filtering Threshold = 20, Unsmoothed) (continued)

It is difficult to make any general statements about the relative performances of the average linkage and k-means clustering results in Figure 2.11. For a “tight” cluster, the spread of the lines representing the gene profiles should be small. Clusters which do not exhibit such patterns are less useful and may contain a high degree of noise.

The second cluster analysis is performed on the smoothed data using a threshold value of 20. Table 2.11 gives the number of elements assigned to each cluster by the average linkage and k-means clustering algorithms.

Table 2.11: Cluster Sizes for the Average Linkage and K-means Techniques (Filtering Threshold = 20, Smoothed)

Cluster Number	Average Linkage	K-means
1	40	15
2	20	58
3	20	8
4	20	10
5	16	18
6	4	20
7	20	11

Figure 2.12 contains plots of the seven clusters for each method. Since cluster labeling is arbitrary, the most similar clusters are usually matched by visual inspection. However, the cluster results for the smoothed data are so different that matching them is difficult. Thus, the clusters are presented using the labels assigned by the two algorithms.

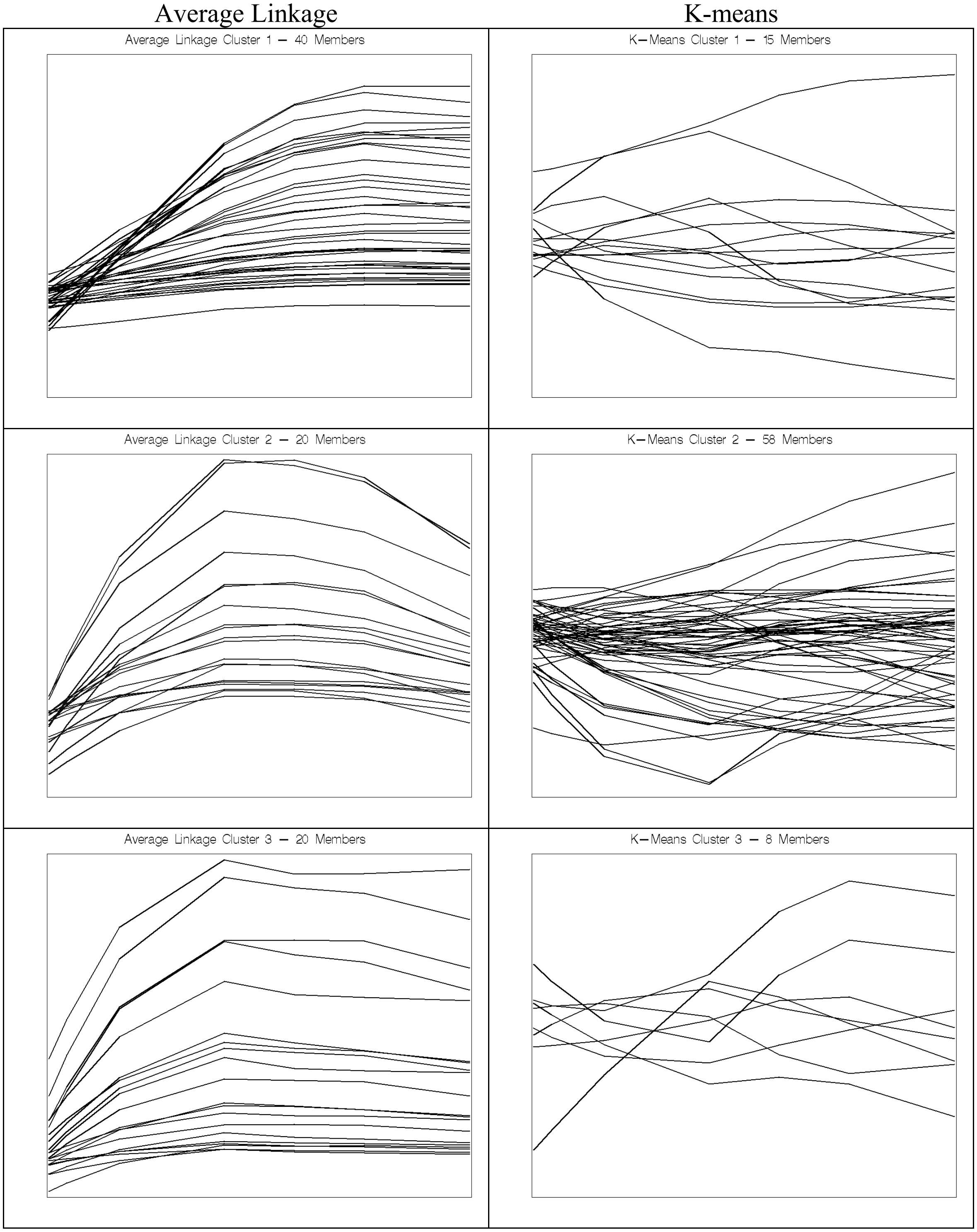


Figure 2.12: Average Linkage and K-Means Clusters
(Filtering Threshold = 20, Smoothed)

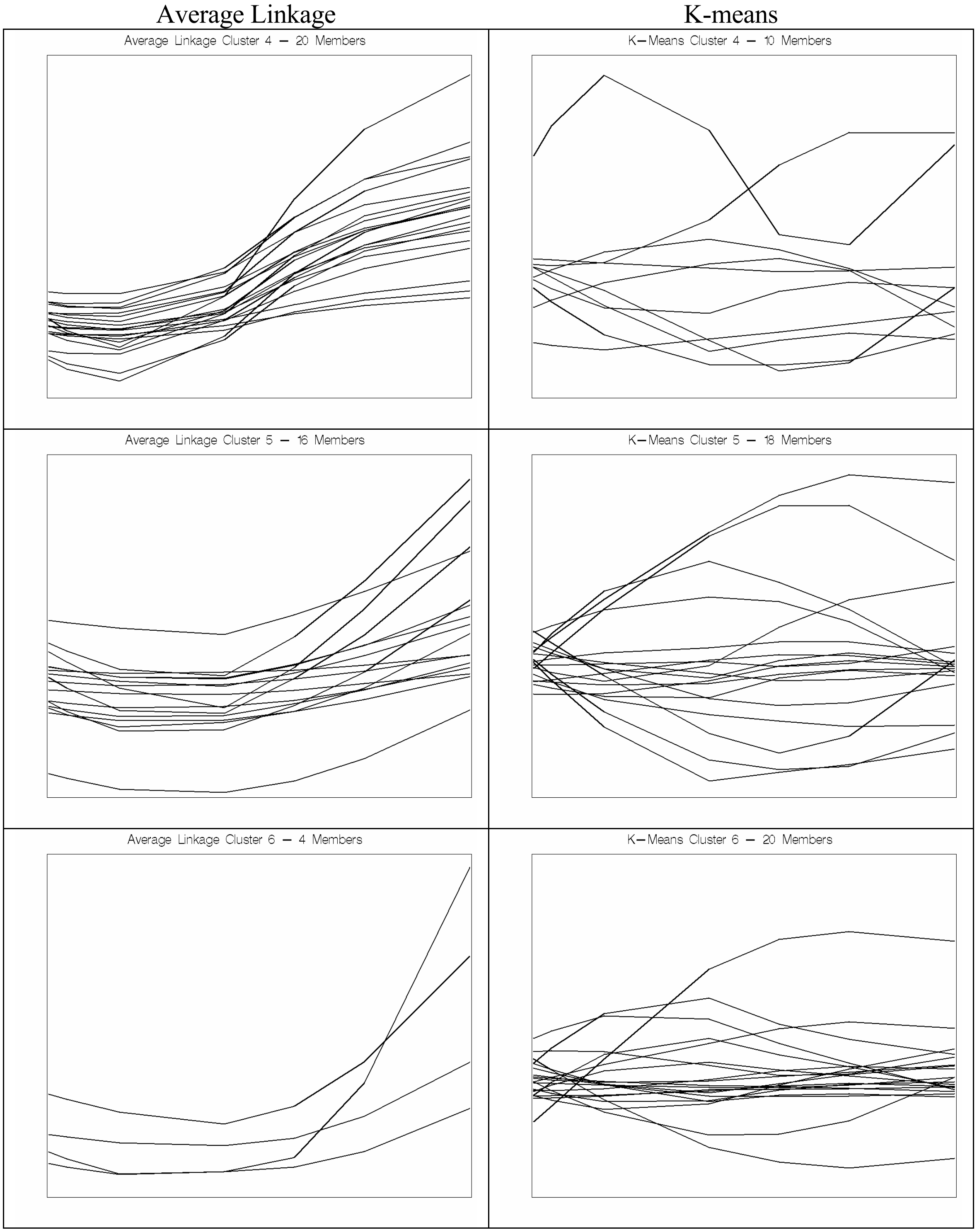


Figure 2.12: Average Linkage and K-Means Clusters
(Filtering Threshold = 20, Smoothed)
(continued)

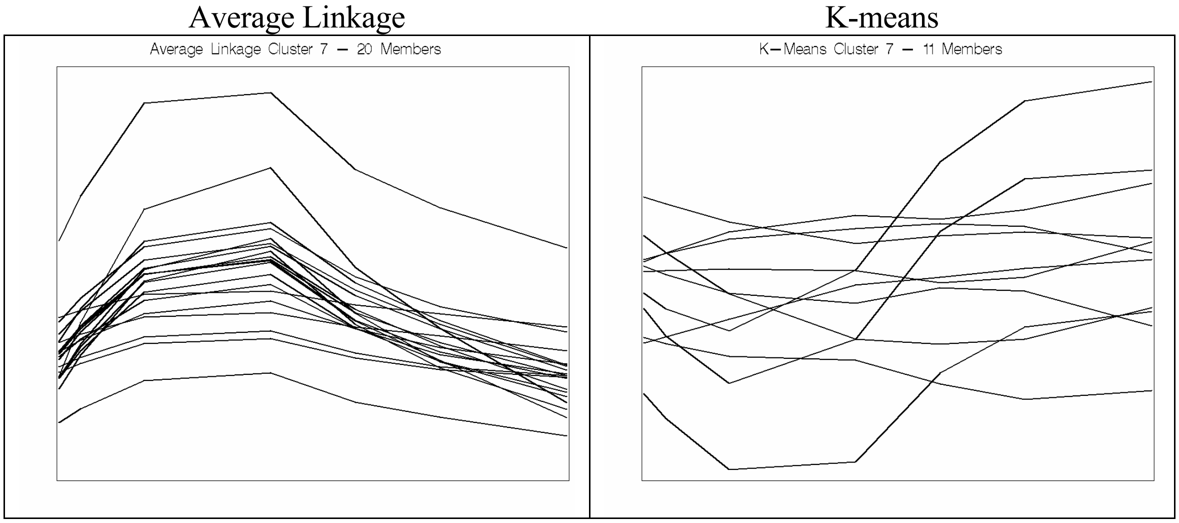


Figure 2.12: Average Linkage and K-Means Clusters
(Filtering Threshold = 20, Smoothed)
(continued)

It is difficult to make any general statements about the relative performances of the average linkage and k-means clustering results in Figure 2.12. However, the average linkage clusters do look a little "tighter", which indicates that this method may perform better for clustering genes having similar profiles for these smoothed data.

The third cluster analysis is performed on the unsmoothed data using a threshold value of 100. Table 2.12 gives the number of elements assigned to each cluster by the average linkage and k-means clustering algorithms.

Table 2.12: Cluster Sizes for the Average Linkage and K-means Techniques
(Filtering Threshold = 100, Unsmoothed)

Cluster Number	Average Linkage	K-means
1	155	127
2	106	50
3	141	88
4	52	119
5	110	27
6	89	210
7	13	45

Figure 2.13 contains plots of the seven clusters for each method. Since cluster labeling is arbitrary, an attempt is made to place the most similar clusters (by visual inspection) from each method side by side. This matching process is inexact. Alternative approaches for comparing clusters are discussed in Chapter 3.

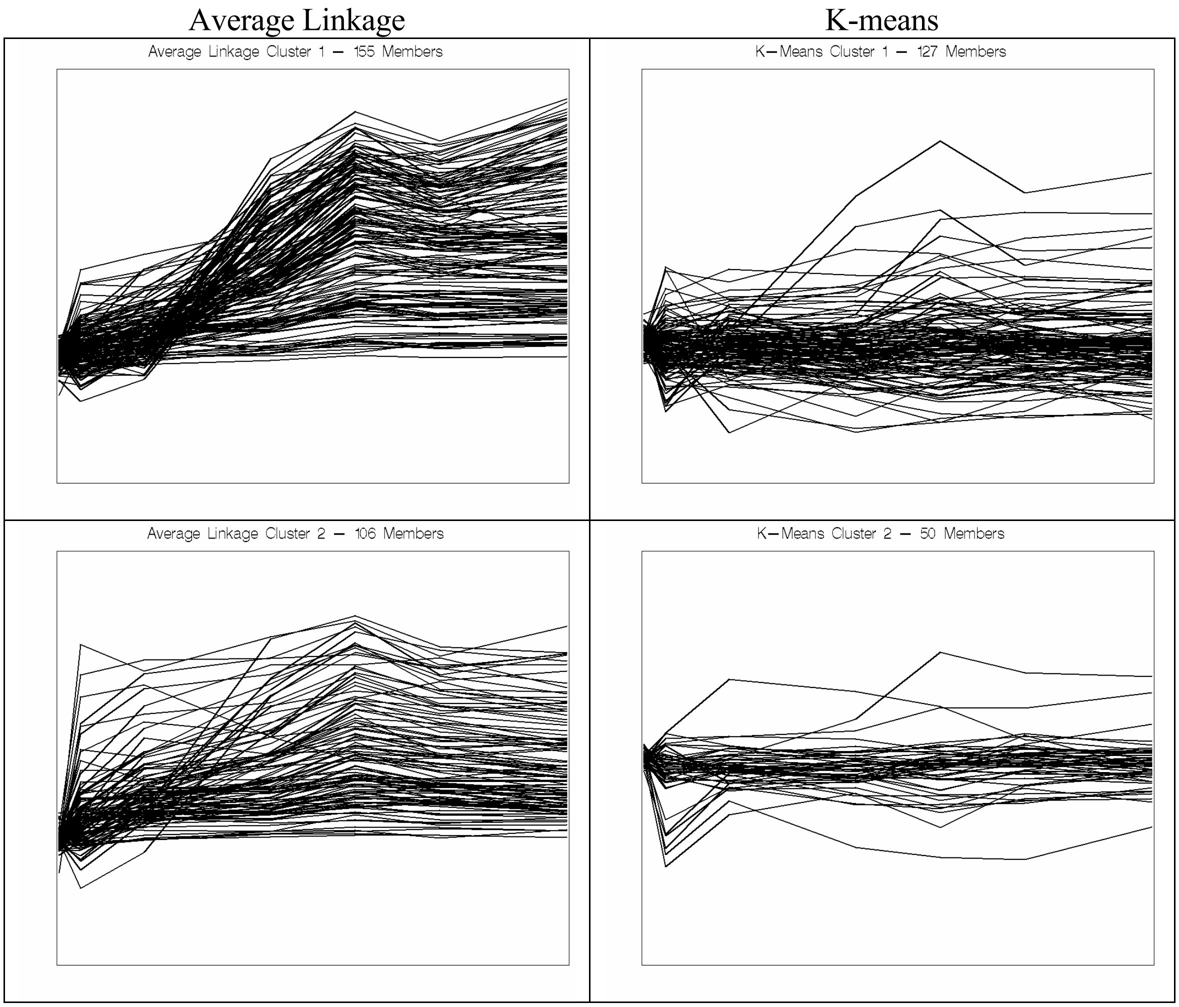


Figure 2.13: Average Linkage and K-Means Clusters
(Filtering Threshold = 100, Unsmoothed)

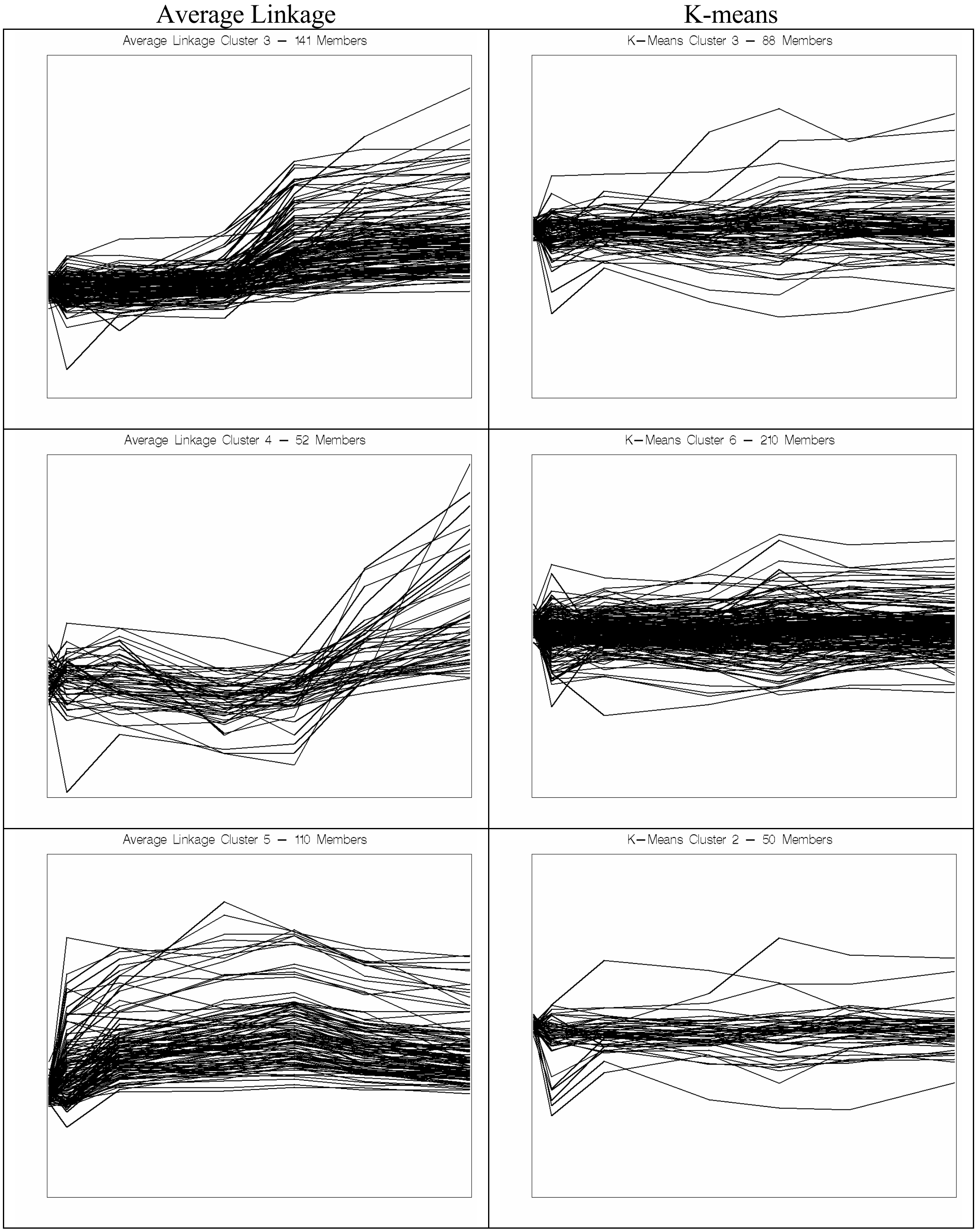


Figure 2.13: Average Linkage and K-Means Clusters
(Filtering Threshold = 100, Unsmoothed)
(continued)

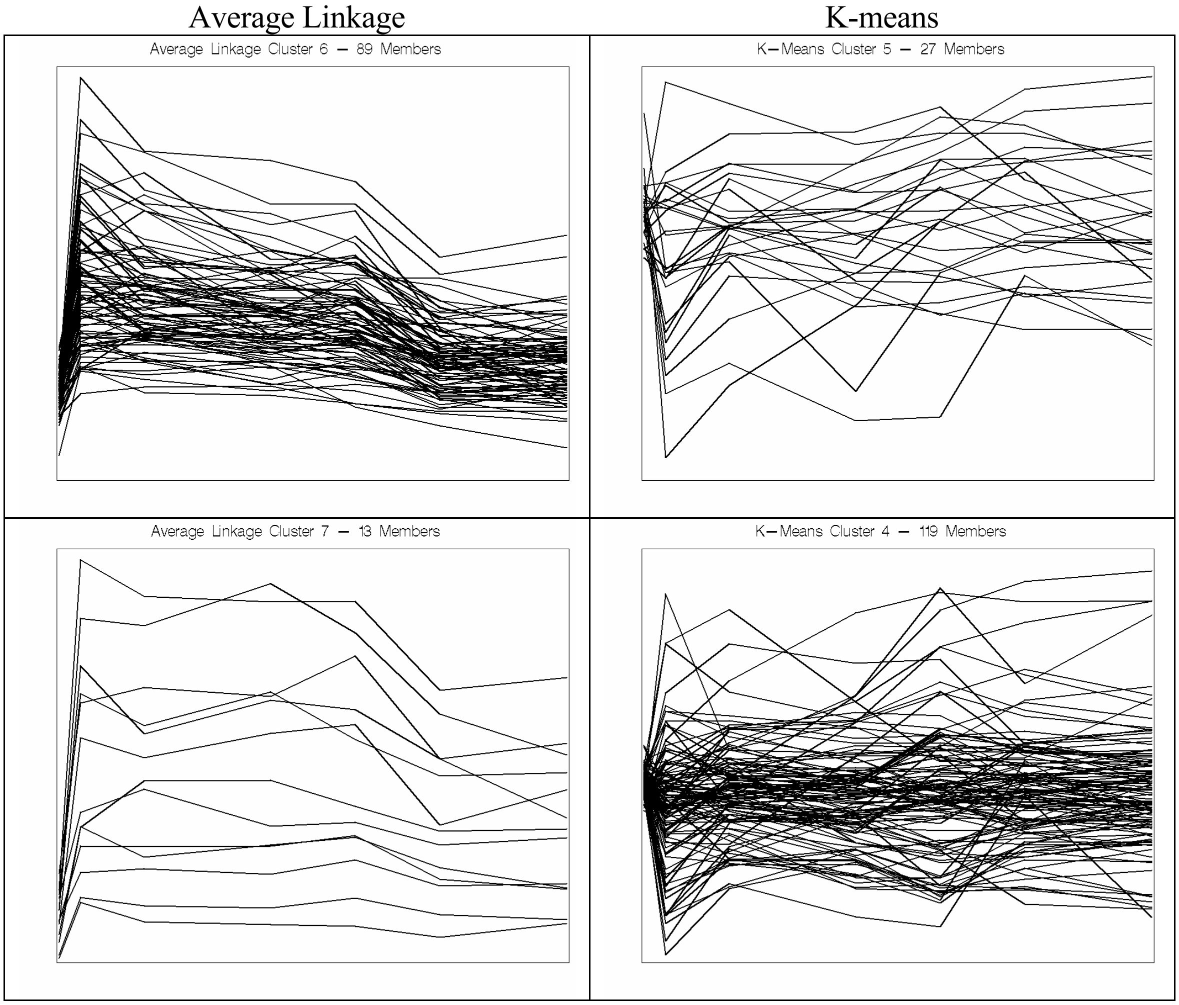


Figure 2.13: Average Linkage and K-Means Clusters
(Filtering Threshold = 100, Unsmoothed)
(continued)

It is difficult to make any general statements about the relative performances of the average linkage and k-means clustering results in Figure 2.13. Some of the cluster plots are obscured due to the large number of observations present. There are a few clusters that look quite noisy (k-means clusters 4 and 5). The more stringent filtering method presented earlier removes some of this “chatter”.

The fourth and final cluster analysis is performed on the smoothed data using a threshold value of 100. Table 2.13 gives the number of elements assigned to each cluster by the average linkage and k-means clustering algorithms.

Table 2.13: Cluster Sizes for the Average Linkage and K-means Techniques
(Filtering Threshold = 100, Smoothed)

Cluster Number	Average Linkage	K-means
1	71	145
2	29	107
3	100	54
4	152	160
5	166	28
6	1	57
7	100	68

Figure 2.14 contains plots of the seven clusters for each method. Since cluster labeling is arbitrary, the most similar clusters are usually matched by visual inspection. However, the cluster results for the smoothed data are so different that matching them is difficult. Thus, the clusters are presented using the labels assigned by the two algorithms.

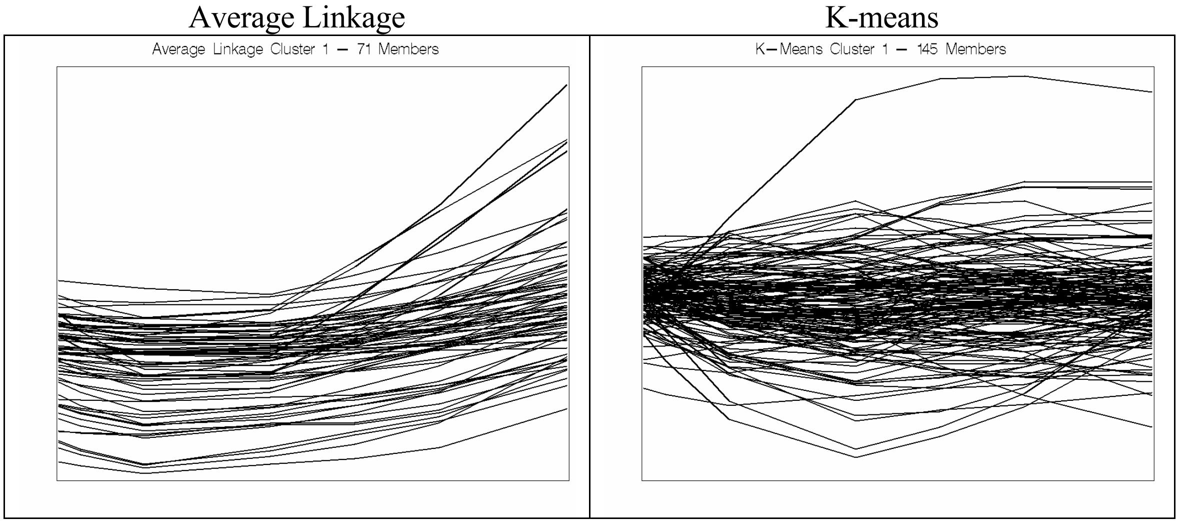


Figure 2.14: Average Linkage and K-Means Clusters
(Filtering Threshold = 100, Smoothed)

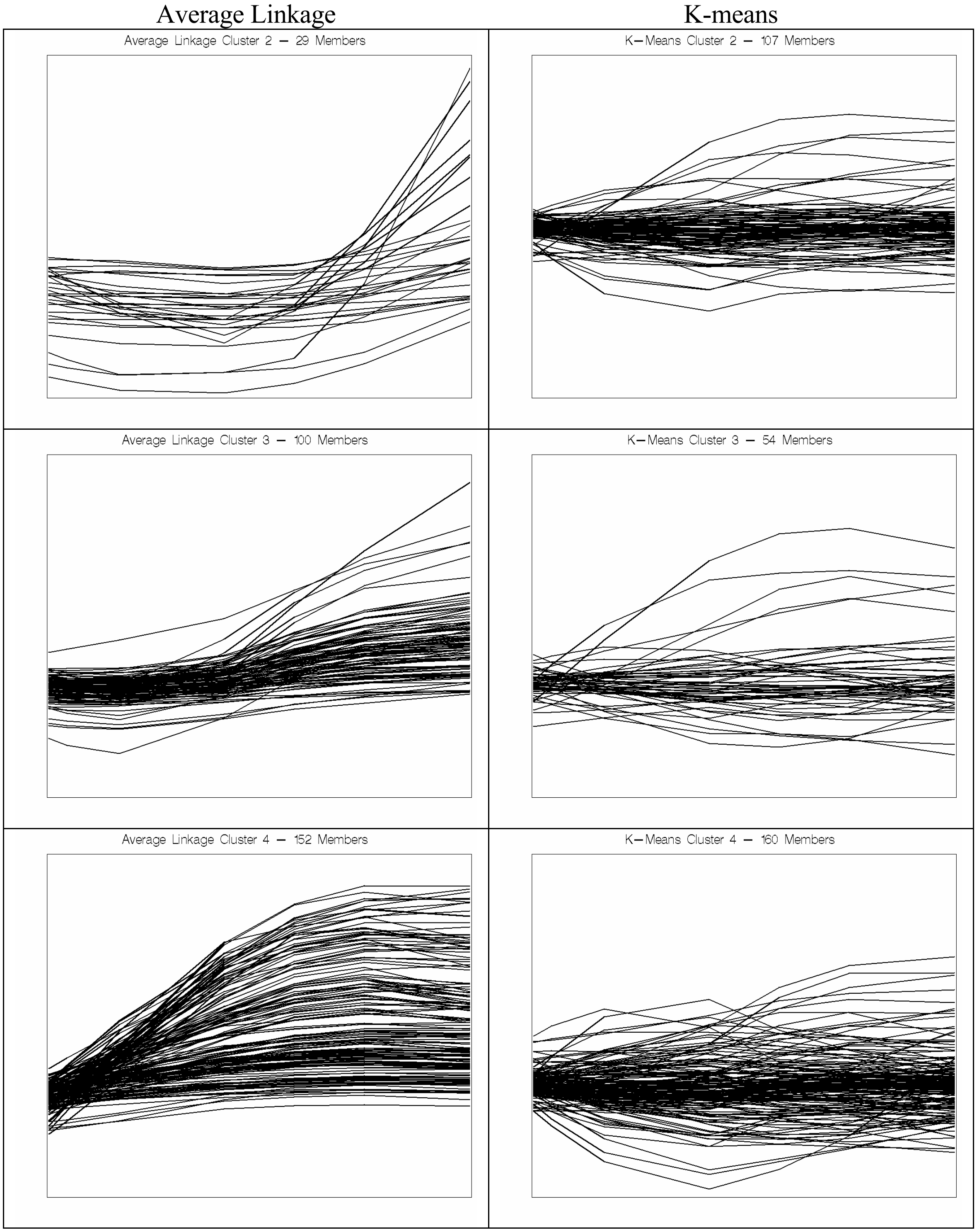


Figure 2.14: Average Linkage and K-Means Clusters
(Filtering Threshold = 100, Smoothed)
(continued)

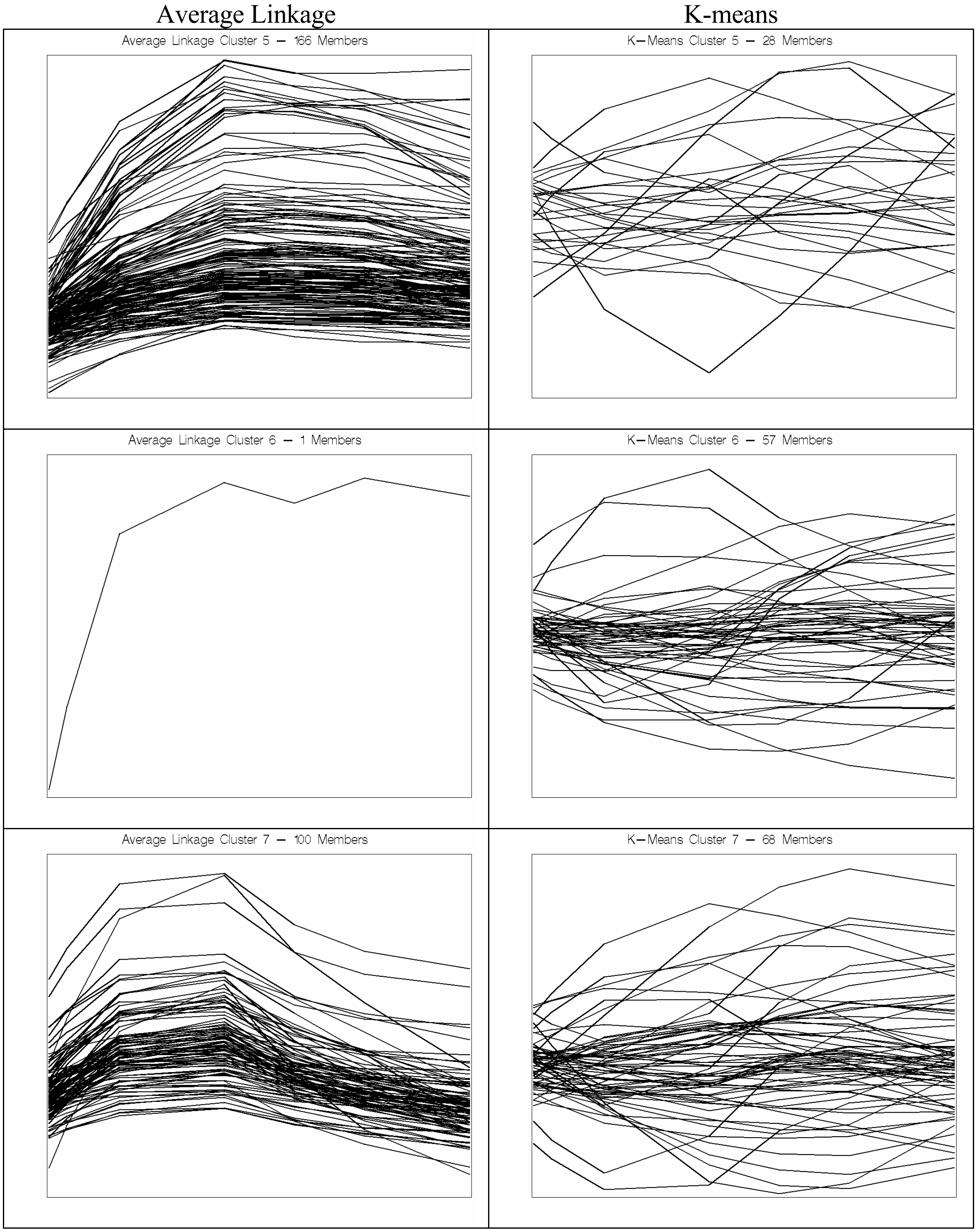


Figure 2.14: Average Linkage and K-Means Clusters
(Filtering Threshold = 100, Smoothed)
(continued)

It is difficult to make any general statements about the relative performances of the average linkage and k-means clustering results in Figure 2.14. However, the average linkage clusters do look a little “tighter”, which indicates that this method may perform better for these data in clustering genes having similar profiles. K-means clusters 5, 6, and 7 are particularly noisy. Average linkage cluster 6 contains only one gene and thus is not useful.

For the cluster analyses performed using the smoothed data, the average linkage clustering algorithm appears to do a better job at grouping genes having similar profiles. However, no clustering algorithm is optimal for all situations, as the results are data dependent.

The more stringent filtering cutoff value of 20 resulted in “tighter” cluster plots for both the smoothed and the unsmoothed data. This lends credence to the filtering algorithm chosen, as the less stringent cutoff value of 100 results in clusters containing more “chatter”. The cutoff values of 20 and 100 were arbitrary and one could examine the cluster profile plots to see how much “chatter” a given cutoff value adds to the plots.

2.9 Conclusion

There are a number of choices to be made when applying non-parametric clustering techniques. A normalization technique may be applied. Normalization is particularly useful in the case of microarray data. One must decide whether to smooth the data or not. Smoothing is only useful for time series data, as categorical data is not suitable for such transformations. Smoothing seems to help in some situations but is quite data dependent. Applying a filtering technique may be helpful. There are a number of ways to filter the data, as discussed in Section 2.7. Finally, to cluster the data, the user must select an appropriate distance measure and clustering technique.

Since there are so many ways to approach the preprocessing and clustering of data, researchers often perform analyses in several different ways and compare the results. However, as discussed in Section 2.8, there is difficulty in comparing cluster solutions from different methods due to the arbitrariness of the cluster labeling. This issue is discussed more fully in Chapter 3.

One of the limitations of non-parametric clustering is that there is no good way to measure how well a clustering result partitions the data. Thus, there is not an optimal way to select a method which clusters the data into the "best" groups. One common approach to this problem is to try to select a method which minimizes the within cluster variances and maximizes the between cluster variances. Another possibility is to apply a parametric clustering technique, which requires making certain assumptions about the distribution of the data. The advantage of using a parametric clustering technique is that statistically based criteria are available for evaluating how well the model is clustering the data. Parametric clustering techniques are discussed in Chapters 4 and 5.

Chapter 3

Comparing Clustering Methods

3.1 Introduction

Comparing cluster analysis solutions is necessary in order to evaluate clusters from different clustering methods and to provide some insight into cluster robustness. Journal articles often list the number of elements contained in each cluster for various clustering methods or present cluster profile graphs without comparing the actual cluster members. One of the reasons for this may be that it is difficult to assign cluster labels to clusters coming from different clustering techniques. In order to compare two competing clustering methods which cluster the data into identical numbers of clusters, one might consider using standard comparison methods for tabulated data, such as the kappa statistic. However, a complication arises due to the fact that the cluster labels are arbitrary. Furthermore, if two clustering methods yield different numbers of clusters, some of these comparison techniques cannot be used because they require a square frequency table. For example, the kappa and the adjusted kappa statistics require square tables. Everitt *et al.* (2001) suggest that if the number of clusters is the same and the cluster agreement is good, the correspondence of labels for the two clustering methods is usually obvious from inspection. However, manual inspection is not feasible for large numbers of clusters or observations. There is no guarantee that the two clusters being compared are the ones that can be reasonably expected to pair with each other.

For the data in this chapter, the true clusters are known. This information allows comparisons to be made between the cluster results and the actual clusters. A measure of cluster effectiveness is obtained by performing these comparisons. However, the true clusters are unknown in most analyses and no clear measure of the success of the clustering can easily be ascertained. Cluster comparisons for these cases is subjective and largely based on the biologically plausibility of the cluster results. One could compare the effectiveness of clustering procedures by examining the within or the between cluster variability. The within cluster variability should be minimal while the between cluster variability should be larger and proportional to the degree of separation between the clusters.

Methods for comparing clusters are the focus of this chapter. First, the data used in the examples is introduced. Next, several methods for assigning cluster labels are described. Various cluster agreement measures are proposed. Examples are given using data for which the actual cluster results are known. The average linkage and k-means clustering methods are used along with the Euclidean distance and Pearson correlation measures. (Chapter 2 contains details on clustering methods and distance measures.) Finally, data from three independently performed microarray experiments targeting the same cell lines are compared. The results are briefly discussed.

3.2 Microarray Data Used in Examples

The National Cancer Institute's (NCI) Developmental Therapeutics Program (DTP) has intensively studied 60 cancer cell lines (Ross *et al.*, 2000), which are known as the NCI 60. This chapter compares the results of three independent microarray experiments which studied the gene expression patterns in 60 human cancer cell lines derived from the nine tumor types listed in Table 3.1.

Table 3.1: Nine Tumor Types from NCI 60

Tumor Type	Number of Cell Lines
Breast Cancer	8
Central Nervous System Cancer	6
Colon Cancer	7
Leukemia	6
Melanoma	8
Non-Small Cell Lung Cancer	9
Ovarian Cancer	6
Prostate Cancer	2
Renal Cancer	8

Each of the three experiments was performed by a different group and targeted the same 60 cell lines. However, the experiments included different numbers of genes and use different microarray technologies. A large number of these genes should be common to the three experiments.

The first data set is from a microarray experiment performed by Ross *et al.* (2000). Using the two color Complementary DNA (cDNA) design, microarrays were prepared by robotically spotting 9,703 human cDNAs on glass microscope slides. The cDNAs included approximately 8,000 unique genes. Each hybridization compared Cy5 labeled cDNA reverse transcribed from mRNA isolated from one of the cell lines with Cy3 labeled cDNA reverse transcribed from a reference mRNA sample. The reference sample, used in all of the hybridizations, was prepared by combining an equal mixture of mRNA from 12 of the cell lines. Only 6,165 genes had complete data for all 60 cell lines. These values were transformed using the usual log background corrected ratio for the two channels. The investigators presented the results from an average linkage cluster analysis using Pearson's correlation as the similarity measure. They found that the cell lines with common tissues of origin tended to cluster together. Cluster analyses were repeated using different subsets of genes to assess cluster robustness. The authors concluded that the clusters appear to be reasonably robust. A major goal of this experiment was to examine the chemosensitivity of the NCI 60 to about 70,000 different chemical compounds. The chemosensitivity data has been analyzed by Ross *et al.* (2000) as well as in separate studies by Paull *et al.* (1989), van Osdol *et al.* (1994), and Weinstein *et al.* (1992, 1997) and is not discussed in this chapter.

The second experiment used the Affymetrix design and examined 5,611 genes for each of the 60 cell lines. This experiment was performed by the Millenium Pharmaceutical Company (http://dtp.nci.nih.gov/mtargets/millenium.html). Poly-A RNA was purified from the 60 human tumor cell lines using the Invitrogen Fast Track 2.0 System. All other steps in RNA extraction and preparation for hybridization were performed as suggested by Wodicka *et al.* (1997). The Affymetrix GeneChip system was used in these experiments. The Hu6000 chip design was used, consisting of 65,000 features each containing on the order of 10 million oligonucleotides designed on the basis of sequence data available from GenBank. The oligonucleotides on the arrays were designed at Affymetrix to cover the complementary strand at the 3' end of the human genes. About 4,000 known fully sequenced human gene cDNA's and more than 2,000 human EST's displaying some similarity with known genes characterized in other organisms are represented in a set of four chips. Most genes are represented by 20 overlapping oligonucleotides. A homosubstitution mismatch oligonucleotide is included for each probe design. The sequence of the oligonucleotide probes on the arrays was selected based on a combination of sequence uniqueness criteria and empirical rules developed at Affymetrix for the selection of oligonucleotides. A quantitative scan of an array and the analysis was done using the Microarray Suite 4.0 software from Affymetrix as described by Wodicka *et al.* (1997). The values reported by the authors are the average of the differences (signal from perfect match - signal from mismatch) after discarding the maximum, the minimum, and any outliers beyond three standard deviations from the mean for the perfect match oligonucleotides. Values less than zero represent measurements for which the mismatched oligonucleotide gave a greater signal than the perfect match oligonucleotide. Clustering results using the average linkage method and Pearson correlation measures of similarity were reported. The investigators found that the cell lines with common tissues of origin tended to cluster together.

The third experiment also used the Affymetrix design and collected data from 7,129 genes for each of the 60 cell lines. This experiment was reported by a group at the Massachusetts Institute of Technology (Staunton *et al.*, 2001). Poly-A selected RNA from each cell line was used to prepare biotinylated cRNA targets. These targets were hybridized to Affymetrix high density Hu6800 microarrays, washed, stained with phycoerythrin conjugated streptavidin, and signal amplified using biotinylated anti-streptavidin antibodies. Expression values were calculated using Affymetrix's Microarray Suite 4.0 software. An expression level of 100 units was assigned to measurements of <100. Setting the threshold in this manner could create a systematic artifactual bias in the distribution of the signals. The authors reported results from an average linkage cluster analysis and found that the cell lines with similar tissues of origin tended to cluster together. Most of Staunton's (2001) paper focuses on chemosensitivty data, which is not discussed in this chapter.

Notice that the three experiments involve different numbers of genes. There is not enough information in the publicly available data files to match up the common genes in the experiments. However, the 60 cell lines are easily matched. To our knowledge, no systematic study of the effect of cluster labeling on clustering method agreement measures has been reported for any of the three experiments. This chapter compares the labeling effects on clusters of cell lines.

3.3 Assigning Cluster Labels

Several approaches for assigning cluster labels are described below. This list is not exhaustive. Our goal is not to advocate any specific method but instead is to encourage consistency in cluster labeling and awareness of the effect of cluster labeling on the interpretability of cluster comparisons. Clustan Graphics version 5.26 was used to perform all of the clustering in this chapter (Wishart, 1999). Data management was performed using SAS version 8.02 (SAS Institute, 1999). The data were not filtered for this analysis.

3.3.1 Naïve Approach to Cluster Labeling

This is the simplest approach for labeling clusters. For two clustering methods A and B, which cluster the data into C_A and C_B clusters, respectively, the cluster labels ($1,\ldots,C_A$ and $1,\ldots,C_B$) arbitrarily assigned by the clustering algorithms are kept. No attempt is made to match the cluster labels based on cluster membership or cluster size.

The following example uses the naïve approach to assign labels to clusters from the Ross *et al.* (2000) NCI 60 microarray data presented in Section 3.2. The k-means clustering algorithm is used (see Chapter 2 for details) to cluster the 60 cell lines into 9 known tumor types. The "real" clusters are known and are referred to as the gold standard. They are compared to the cluster results from the k-means algorithm using the Pearson correlation similarity measure. Table 3.2 summarizes the agreement between the gold standard and the k-means clustering results using existing cluster labels. Note that only 2 out of 9 entries on the diagonal (bolded) are non-zero, indicating that there is poor agreement between the k-means clustering algorithm and the gold standard. Both of the non-zero entries are 1, which indicates only weak agreement between the clustering methods for these two clusters.

Table 3.2: Naïve Cluster Labeling

		Gold Standard								
		1	2	3	4	5	6	7	8	9
k-means	1	**1**	0	2	0	0	0	0	0	0
	2	0	**0**	0	0	6	0	0	0	0
	3	4	1	**0**	5	0	0	0	2	0
	4	0	0	2	**0**	0	7	0	0	0
	5	0	0	0	0	**0**	0	1	0	0
	6	0	6	0	0	0	**0**	0	0	0
	7	1	0	0	0	0	0	**0**	0	0
	8	2	0	3	1	0	1	7	**0**	5
	9	1	0	1	0	0	0	0	0	**1**

Clustering software packages assign cluster labels differently. Thus, using a naïve approach to label clusters for comparison offers little consistency. Also, using this approach fails to make use of the knowledge that clusters having a large number of elements in common should most likely share the same label.

The naïve approach to cluster labeling works when the two competing clustering algorithms assign cluster labels in exactly the same order. This rarely happens in practice, particularly when a large number of clusters is present. If the cluster labels do not occur in exactly the same order, the clusters are said to be mislabeled. The consequences of mislabeling clusters vary in severity and may cause the reported agreement between methods to be weaker than it actually is.

3.3.2 Ranked Approach to Cluster Labeling

The ranked approach to labeling clusters assigns cluster labels based on the ranks of the cluster sizes. For comparing the gold standard with the k-means clustering algorithm (using the Pearson correlation similarity measure), Table 3.3 gives the ranked list of clusters along with the label assigned.

Table 3.3: Ranked List of Cluster Members with Labels Assigned

	Number of Cluster Members								
Gold Standard	2	6	6	6	7	8	8	8	9
k-means	1	1	3	3	6	6	9	12	19
Cluster Label Assigned	1	2	3	4	5	6	7	8	9

The number of cluster members in the k-means clusters do not correspond well with the number of cluster members in the gold standard. Also, several clusters have the same number of members in both the gold standard and k-means cluster results. These tied values are a potential weakness of the ranked approach to labeling clusters, as there is no way to establish a proper ranking for tied cluster sizes. For example, the clusters labeled 3 and 4 are tied in terms of the number of cluster members for both the gold standard and k-means clustering results. The gold standard has 6 members for each of these clusters while the k-means results indicate only 3 members for these two clusters. Suppose we call these cluster pairings 6_3, 6_4, 3_3, and 3_4, where the subscript represents the cluster label, the 6’s come from the gold standard, and the 3’s come from the k-means clustering results. Currently, the cluster pair $(6_3,3_3)$ is assigned cluster label 3 and the cluster pair $(6_4,3_4)$ is assigned label 4. An equally valid possibility would be to assign the cluster pair $(6_3,3_4)$ to label 3 and $(6_4,3_3)$ to label 4 (or vice versa). It is possible to develop an algorithm which assigns cluster labels based on all of the possible combinations of tied clusters and chooses the labeling scheme offering the best cluster agreement. However, such an algorithm is difficult to implement and the computational resources required would quickly increase with the number of clusters.

Table 3.4 summarizes the agreement between the gold standard and k-means clustering results using the new rank based cluster labels.

Table 3.4: Rank Based Cluster Labeling

		Gold Standard								
		1	2	3	4	5	6	7	8	9
k-means	1	**0**	0	0	0	0	0	0	1	0
	2	0	**0**	0	0	0	0	0	0	1
	3	0	0	**0**	0	0	2	0	0	1
	4	0	0	0	**1**	0	1	0	0	1
	5	0	0	6	0	**0**	0	0	0	0
	6	0	0	0	0	6	**0**	0	0	0
	7	0	0	0	0	0	0	**7**	0	0
	8	2	5	0	0	1	0	0	**0**	4
	9	0	1	0	5	0	3	1	7	**2**

Note that only 3 out of 9 entries on the diagonal (bolded) are non-zero, indicating that there is poor agreement between the k-means clustering algorithm and the gold standard.

Observe that 2 out of 3 of the diagonal entries have values of 2 or less, indicating weak agreement between the two clustering methods. However, the agreement for the ranked cluster labeling approach is better than that for the naïve cluster labeling approach.

3.3.3 Best Case Approach to Cluster Labeling

The best case cluster labeling approach assigns cluster labels based on the maximum number of elements that the clusters from two different methods have in common. This approach is called the best case approach because it assumes that clusters sharing the maximum number of common members should receive the same label. The algorithm iterates until all of the clusters are assigned a label. This procedure is illustrated in the example below. Table 3.5 shows how well the clusters agree based on the cluster labels assigned by the clustering algorithms. The k-means clustering algorithm uses the Pearson correlation similarity measure.

Table 3.5: Agreement between Arbitrarily Assigned Cluster Labels

		Gold Standard								
		1	2	3	4	5	6	7	8	9
k-means	1	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0
	3	4	1	0	0	0	0	0	0	0
	4	0	0	2	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0
	6	0	6	0	0	0	0	0	0	0
	7	1	0	0	0	0	0	0	0	0
	8	2	0	3	1	0	1	7	0	0
	9	1	0	1	0	0	0	0	0	1

The initial step in the best case cluster labeling algorithm is to find the maximum number of matches in the first column. The number circled, 4, in Table 3.5 is the maximum.

This indicates that k-means cluster 3 and gold standard cluster 1 receive the same cluster label, 1, because they have the most elements in common. Since they are now assigned cluster labels, k-means cluster 3 and gold standard cluster 1 are eliminated from consideration, thus forming Table 3.6.

Table 3.6: Best Case Cluster Label Assignment, 2nd Iteration

		Gold Standard							
		2	3	4	5	6	7	8	9
k-means	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0
	4	0	2	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0
	6	6	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0
	8	0	3	1	0	1	7	0	0
	9	0	1	0	0	0	0	0	1

The process is repeated, with 6 being the maximum entry in the first column of Table 3.6. This indicates that k-means cluster 6 and gold standard cluster 2 receive the same cluster label, 2, because they have the most elements in common.

This process iterates until all of the k-means clusters are assigned new labels. Table 3.7 summarizes the agreement between the gold standard and k-means clustering results using the new cluster labels.

Table 3.7: Best Case Approach to Cluster Labeling

		Gold Standard								
		1	2	3	4	5	6	7	8	9
k-means	1	**4**	1	0	5	0	0	0	2	0
	2	0	**6**	0	0	0	0	0	0	0
	3	2	0	**3**	1	0	1	7	0	5
	4	1	0	2	**0**	0	0	0	0	0
	5	0	0	0	0	**6**	0	0	0	0
	6	0	0	2	0	0	**7**	0	0	0
	7	0	0	0	0	0	0	**1**	0	0
	8	1	0	0	0	0	0	0	**0**	0
	9	1	0	1	0	0	0	0	0	**1**

Note that now 7 out of 9 entries on the diagonal (bolded) are non-zero, indicating much improved cluster agreement between the k-means clustering algorithm and the gold standard.

The best case cluster labeling algorithm is limited in that there could be ties on the number of members shared between clusters. Tied values should be treated consistently. In the event of a tie, this algorithm assigns cluster labels based on the first pair of tied clusters. Ties are discussed in more detail in Section 3.3.2.

The best case cluster labeling algorithm assigns cluster labels on the basis on the maximum number of common members. A possible improvement would be to use the percentage of common members for cluster labeling instead of a simple count. Such a labeling scheme could prevent some undesirable situations, such as a large cluster in one clustering method being assigned the same label as a much smaller cluster from another clustering method. This could occur when the clustering results for the large cluster agree poorly but the results for the small cluster agree well and both cluster pairs have a similar count of observations for which the clusters agree.

3.4 Measuring Cluster Agreement

Once the cluster labels are assigned and the data is reduced to a two dimensional frequency table, methods for measuring how well the two clustering algorithms agree are needed. Ideally, these measures should not be influenced by the number of observations or clusters present. If both clustering methods agree strongly, one would expect more observations to fall on the diagonal. Thus, the agreement measure should reflect how many off-diagonal observations are present. Several agreement measures are proposed and discussed below.

3.4.1 Kappa and Weighted Kappa Statistics

The kappa statistic is a widely used measure of how well two frequency tables agree. Cohen (1960) first introduced the kappa statistic. This statistic is introduced using the symbolic data presented in Table 3.8.

Table 3.8: Example for the Kappa Statistic Presentation

		Cluster Method 2				Total
		1	2	3	4	
Cluster Method 1	1	O_{11}	O_{12}	O_{13}	O_{14}	R_1
	2	O_{21}	O_{22}	O_{23}	O_{24}	R_2
	3	O_{31}	O_{32}	O_{33}	O_{34}	R_3
	4	O_{41}	O_{42}	O_{43}	O_{44}	R_4
Total		C_1	C_2	C_3	C_4	n

Both the observed proportion of observations that agree and the expected proportion of observations that should agree are calculated. The observed proportion that agree, P_o, is the sum of the observed diagonal elements in the frequency table divided by the total number of observations classified. Thus, $P_o = \sum_{i=1}^{c} O_{ii} / n$, where c is the number of columns. Similarly, the expected proportion that agree is: $P_e = \sum_{i=1}^{c} R_i C_i / n^2$. The kappa statistic is calculated as:

$$\hat{\kappa} = \frac{P_o - P_e}{1 - P_e}. \tag{3.1}$$

The kappa statistic takes on a value of one when the agreement is perfect and zero if the agreement is the same as that expected by chance. Kappa values less than zero, which occur rarely and are usually interpreted as zero, indicate that the agreement is less than that expected by chance.

The asymptotic variance of the kappa statistic is (Fleiss *et al.*, 1969):

$$\mathrm{var}\left(\hat{\kappa}\right) = \frac{A + B - C}{n\left(1 - P_e\right)^2} \tag{3.2}$$

where $A = \sum_{i=1}^{c} \rho_{ii} \left[1 - \left(\rho_{i.} + \rho_{.i}\right)\left(1 - \hat{\kappa}\right)\right]^2$, $B = \left(1 - \hat{\kappa}\right)^2 \sum_{i=1}^{c} \sum_{j=1}^{c} \rho_{ij} \left(\rho_{.i} + \rho_{j.}\right)^2$, $i \neq j$, $C = \left[\hat{\kappa} - P_e\left(1 - \hat{\kappa}\right)^2\right]$, and the $\rho's$ indicate the proportion of observations falling in a given cell of the table. A confidence interval for the kappa statistic may be calculated as:

$$\hat{\kappa} \pm z_{1-\alpha/2} \sqrt{\mathrm{var}\left(\hat{\kappa}\right)}\ , \tag{3.3}$$

where $z_{1-\alpha/2}$ is the usual $100(1-\alpha/2)$ percentile of the standard normal distribution. For example, to obtain 95 percent confidence limits, $\alpha = 0.05$ and $z_{1-\alpha/2} = 1.96$.

One limitation of the kappa statistic is that it does not apply to tables which have different numbers of rows and columns. A second limitation of the kappa statistic is that it treats all of the off-diagonal observations equally. In many cases, the researcher may prefer to treat observations falling farther from the diagonal differentially.

The weighted kappa statistic generalizes the kappa statistic. The weighted kappa statistic assigns a weight w_{ij} to each cell depending on its location in the frequency table. These weights must conform to the following conditions:

1. $0 \le w_{ij} < 1 \ \forall \ i \ne j$
2. $w_{ii} = 1 \ \forall \ i$
3. $w_{ij} = w_{ji} \ \forall$ i, j

Analogous to the simple kappa statistic, the observed agreement proportion is:

$P_{o(w)} = \sum_{i=1}^{c} \sum_{j=1}^{c} w_{ij} O_{ij} / n$, where c is the number of columns. The expected agreement proportion is: $P_{e(w)} = \sum_{i=1}^{c} \sum_{j=1}^{c} w_{ij} O_{i.} O_{.j} / n^2$. The weighted kappa statistic is calculated as:

$$\hat{\kappa}_w = \frac{P_{o(w)} - P_{e(w)}}{1 - P_{e(w)}} \tag{3.4}$$

The weighted kappa statistic is interpreted in the same way as the simple kappa statistic. The weighted kappa statistic also requires square frequency tables.

The asymptotic variance of the weighted kappa statistic is (Fleiss *et al.*, 1969):

$$\operatorname{var}\left(\hat{\kappa}_w\right) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} \rho_{ij}\left[w_{ij} - \left(\overline{w}_{i.} + \overline{w}_{.j}\right)\left(1-\hat{\kappa}_w\right)\right]^2 - \left[\hat{\kappa}_w - P_{e(w)}\left(1-\hat{\kappa}_w\right)\right]^2}{n\left(1-P_{e(w)}\right)^2} \tag{3.5}$$

where $\overline{w}_{i.} = \sum_{j=1}^{c} \rho_{.j} w_{ij}$ and $\overline{w}_{.j} = \sum_{i=1}^{c} \rho_{i.} w_{ij}$. A confidence interval for the kappa statistic may be calculated as:

$$\hat{\kappa}_w \pm z_{1-\alpha/2}\sqrt{\operatorname{var}\left(\hat{\kappa}_w\right)} \tag{3.6}$$

where $z_{1-\alpha/2}$ is the usual $100(1-\alpha/2)$ percentile of the standard normal distribution. For example, to obtain 95 percent confidence limits, $\alpha = 0.05$ and $z_{1-\alpha/2} = 1.96$.

There are many choices for assigning weights. Two of the most commonly used methods for assigning weights are the Cicchetti-Allison and Fleiss-Cohen methods. The weighted kappa statistics calculated in this chapter use the Cicchetti-Allison weights.

Cicchetti and Allison (1971) proposed one of the first broadly accepted weighting schemes. Let *c* be the number of categories and C_i and C_j represent the column totals for columns *i* and *j*, respectively. The Cicchetti-Allison weights are calculated as:

$$w_{ij}^{CA} = 1 - \frac{\left|C_i - C_j\right|}{C_c - C_1} \tag{3.7}$$

Fleiss and Cohen (1973) proposed what became known as the Fleiss-Cohen weighting scheme. The Fleiss-Cohen weights are calculated as:

$$w_{ij}^{FC} = 1 - \frac{\left(C_i - C_j\right)^2}{\left(C_c - C_1\right)^2} \tag{3.8}$$

Both of these weighting schemes assign more weight for frequency table entries closer to the diagonal and less weight for those falling farther from the diagonal.

3.4.2 Rand Index

The Rand index (Rand, 1971) is a frequency table agreement measure based on the pairwise comparison of n observations rather than a simple cross tabulation of the frequencies. The index can be used to compare frequency tables having different numbers of rows and columns. The Rand index computes the proportion of the total of ${}_nC_2$ observation pairs that agree (Everitt *et al.*, 2001). Agreement means that either both of the observations in the pair fall into the same cluster according to both partitions or both observations fall into different clusters according to both partitions. The Rand index is defined as:

$$I_R = \frac{A}{\binom{n}{2}} \tag{3.9}$$

where $A = \binom{n}{2} + 2\sum_{i=1}^{c1}\sum_{j=1}^{c2} O_{ij}^2 - \left(\sum_{i=1}^{c1} O_{i.}^2 + \sum_{j=1}^{c2} O_{.j}^2\right)$, where $O_{i.} = \sum_{i=1}^{c1} O_{ij}$ and $\mathrm{O}_{.j} = \sum_{j=1}^{c2} O_{ij}$, n is the total number of observations, and c1 and c2 are the number of rows and columns in the frequency table, respectively.

The derivation of Equation 3.9 comes from the argument presented by Rand (1971). Consider n points, $X_1, X_2, \ldots, X_n$, and two clusters of them, $Y = \{Y_1, Y_2, \ldots, Y_n\}$ and $Y' = \left\{Y_1', Y_2', \ldots, Y_n'\right\}$. A similarity measure, *S*, between the two clusterings of the same data, *Y* and Y', can be defined as $S\left(Y, Y'\right)$, which is equal to the number of cluster assignment pairs in agreement normalized by the total number of pairs. This formula may be written as:

$$S(Y, Y') = \frac{\sum_{i=1}^{c1}\sum_{j=1}^{c2} \gamma_{ij}}{\binom{n}{2}} \tag{3.10}$$

where $i < j$ and:

$$\gamma_{ij} = \begin{cases} 1 \text{ if } \exists\ k \text{ and } k' \ \ni \text{ both } X_i \text{ and } X_j \text{ are in both } Y_k \text{ and } Y_{k'}' \\ 1 \text{ if } \exists\ k \text{ and } k' \ \ni \text{ both } X_i \text{ is in both } Y_k \text{ and } Y_{k'}' \text{ while } X_j \text{ is in neither } Y_k \text{ nor } Y_{k'}' \\ 0 \text{ otherwise} \end{cases}.$$

$S\left(Y, Y'\right)$ may be expressed in the convenient computational form, called the Rand index, given in Equation 3.9. The asymptotic variance for the Rand index is not available in the literature.

The Rand index has an expected value slightly greater than 0 and ranges from slightly less than 0 to 1. If the partitions agree perfectly, the Rand index is 1. The Rand index is a similarity measure, thus (1 – the Rand index) is a distance measure. Fowlkes and Mallows (1983) point out that the Rand index tends to increase as the number of clusters increases and that the possible range of values for the index is quite narrow. The adjusted Rand index (discussed in the next section) compensates for these issues.

The Rand index is used to compare cluster solutions in the microarray literature. For example, Yeung *et al.* (2001) compare the results of running several different clustering algorithms to cluster genes using both real and simulated microarray gene expression data. They use the Rand index as the primary measure of cluster agreement. Yeung and Ruzzo (2001) examined principal component based methods for clustering microarray data. They analyzed data from simulated and empirical microarray experiments. They compared the clustering results using k-means and principle component based methods using the Rand index and the adjusted Rand index as the agreement measures. The clustering methods agreed poorly. Yeung and Ruzzo do not recommend principal component based methods for general use due to the computational resources required.

3.4.3 Adjusted Rand Index

The adjusted Rand index has an expected value of zero and ranges from -1 to 1. Due to its expanded range, the adjusted Rand index is more sensitive than the Rand index. The adjusted Rand index does not increase with the number of clusters. Hubert and Arabie

(1985) developed the adjusted Rand index. The adjusted Rand index, I_{AR}, is analogous to the kappa coefficient since it ranges from chance levels (0) to perfect agreement (1) and measures the agreement over and above that expected by chance (Everitt *et al.*, 2001). Any Rand index value smaller than zero indicates less than chance agreement and is generally set to zero in practice. This index is calculated by:

$I_{AR} = N / D$, where c1 and c2 are as defined previously,

$$N = \sum_{i=1}^{c1} \sum_{j=1}^{c2} \binom{O_{ij}}{2} - \frac{\sum_{i=1}^{c1} \binom{O_{i.}}{2} \sum_{j=1}^{c2} \binom{O_{.j}}{2}}{\binom{n}{2}} \tag{3.11}$$

and

$$D = \frac{\sum_{i=1}^{c1} \binom{O_{i.}}{2} + \sum_{j=1}^{c2} \binom{O_{.j}}{2}}{2} - \frac{\sum_{i=1}^{c1} \binom{O_{i.}}{2} \sum_{j=1}^{c2} \binom{O_{.j}}{2}}{\binom{n}{2}} \tag{3.12}$$

The asymptotic variance for the Rand index is not available in the literature.

Milligan and Cooper (1986) evaluated many different indices for measuring the agreement between two partitions with different numbers of clusters and recommend the adjusted Rand index over any competing measures. The adjusted Rand index typically takes on lower values than the Rand index.

3.4.4 Examples of Comparing Cluster Agreement

This section compares the various cluster agreement measures using the data from Ross *et al.* (2000) presented in Section 3.2. Recall that the 60 tumor cell lines should cluster into 9 known tumor types. Table 3.9 compares the agreement between the gold standard, average linkage, and k-means clustering methods using the Euclidean distance measure. The kappa and adjusted kappa statistics and their 95 percent confidence intervals are calculated using the naïve, ranked, and best case cluster labeling algorithms described in Section 3.3. The Rand and adjusted Rand statistics are also calculated. The SAS code for these analyses is listed in Appendix 3.1.

Table 3.9: Cluster Agreement for the Euclidean Distance Measure

Comparison	**Kappa**			**Weighted Kappa**			**RI**	**ARI**
	N	**R**	**BC**	**N**	**R**	**BC**		
GS vs. AL	0*	0.01	0.18	0*	0*	0.15	0.47	0.04
	(0*,0.01)	(0*,0.09)	(0.08,0.28)	(0*,0.04)	(0*,0.07)	(0.01,0.29)		
GS vs. KM	0*	0.04	0.50	0.13	0.06	0.38	0.84	0.34
	(0*,0.01)	(0*,0.15)	(0.36,0.63)	(0*,0.29)	(0*,0.20)	(0.19,0.57)		
AL vs. KM	0*	0.07	0.25	0*	0.39	0.11	0.59	0.24
	(0*,0.07)	(0.01,0.12)	(0.13,0.38)	(0*,0.01)	(0.25,0.53)	(0*,0.25)		

where RI = Rand Index, ARI = Adjusted Rand Index
GS = gold standard, KM = k-means, AL = average linkage
N = naïve labeling, R = ranked labeling, BC = best case labeling
Numbers in ()'s represent 95% confidence intervals
* = negative calculated value was set to zero

The results shown in Table 3.9 are discussed below. First of all, notice that for the kappa statistic, all three comparisons show better agreement (0.18, 0.50, 0.25) for the best case cluster labeling algorithm than for the ranked and naïve algorithms. This indicates that the best case cluster labeling algorithm may more closely reflect the true cluster similarity than the other labeling algorithms. The weighted kappa statistics vary widely. Focusing on the best case cluster labeling algorithm, we see that both the kappa and the weighted kappa statistics are the highest (0.50 and 0.38, respectively) for the gold standard versus k-means cluster results comparison. The Rand index and the adjusted Rand index for this comparison (RI = 0.84 and ARI = 0.34) are also the maximums for the three comparisons. This evidence suggests that the k-means algorithm does the best at reproducing the actual clusters. If we look at our measures of choice, the Rand index and adjusted Rand index, we see that the average linkage and k-means algorithms agree quite well (RI = 0.59 and ARI = 0.24), while the gold standard and average linkage results agree less strongly (RI = 0.47 and ARI = 0.04). Cluster agreement measures depend heavily on the labeling algorithm employed as well as how readily separable the clusters are.

Each comparison between two clustering methods in Table 3.9 is based on an underlying frequency table. The kappa, weighted kappa, Rand index, and adjusted Rand index cluster agreement measures are calculated using these tables. These 9 x 9 tables for the gold standard versus k-means cluster comparisons reported in the second line of Table 3.9 are given below.

Table 3.10 gives the frequency table for the naïve cluster labeling algorithm using the Euclidean distance measure for the gold standard versus k-means cluster comparison. Notice in Table 3.10 that most of the diagonal entries are zero, indicating weak agreement between the two clustering methods. This weak agreement is reflected in the agreement measures reported in Table 3.9.

Table 3.10: Frequency Table for Gold Standard versus K-means Cluster Comparison using the Naïve Cluster Labeling Algorithm and the Euclidean Distance Measure

		K-means								
		1	2	3	4	5	6	7	8	9
Gold Standard	1	**1**	0	4	0	0	0	1	2	1
	2	0	**0**	1	0	0	6	0	0	0
	3	2	0	**0**	2	0	0	0	3	1
	4	0	0	5	**0**	0	0	0	1	0
	5	0	6	0	0	**0**	0	0	0	0
	6	0	0	0	7	0	**0**	0	1	0
	7	0	0	0	0	1	0	**0**	7	0
	8	0	0	2	0	0	0	0	**0**	0
	9	0	0	0	0	0	0	0	5	**1**

Table 3.11 gives the frequency table for the ranked cluster labeling algorithm using the Euclidean distance measure for the gold standard versus k-means cluster comparison.

Table 3.11: Frequency Table for Gold Standard versus K-means Cluster Comparison using the Ranked Cluster Labeling Algorithm and the Euclidean Distance Measure

		K-means								
		1	2	3	4	5	6	7	8	9
Gold Standard	1	**0**	0	0	0	0	0	0	2	0
	2	0	**0**	0	0	0	0	0	5	1
	3	0	0	**0**	0	6	0	0	0	0
	4	0	0	0	**1**	0	0	0	0	5
	5	0	0	0	0	**0**	6	0	1	0
	6	0	0	2	1	0	**0**	2	0	3
	7	0	0	0	0	0	0	**7**	0	1
	8	1	0	0	0	0	0	0	**0**	7
	9	0	1	1	1	0	0	0	4	**2**

Notice in Table 3.11 that there are more entries falling on the diagonal than there were in Table 3.10, indicating slightly better agreement between the two clustering methods. This stronger agreement is reflected in the agreement measures reported in Table 3.9.

Table 3.12 gives the frequency table for the best case cluster labeling algorithm using the Euclidean distance measure for the gold standard versus k-means cluster comparison.

Table 3.12: Frequency Table for Gold Standard versus K-means Cluster Comparison using the Best Case Cluster Labeling Algorithm and the Euclidean Distance Measure

		K-means								
		1	2	3	4	5	6	7	8	9
Gold Standard	1	**2**	0	0	2	0	0	0	3	1
	2	0	**6**	0	0	0	0	0	0	0
	3	0	0	**5**	0	0	0	0	1	0
	4	0	0	0	**7**	0	0	0	1	0
	5	0	0	0	0	**1**	0	0	7	0
	6	0	0	1	0	0	**6**	0	0	0
	7	1	0	4	0	0	0	**1**	2	1
	8	0	0	0	0	0	0	0	**5**	1
	9	0	0	2	0	0	0	0	0	**0**

Notice in Table 3.12 that there are more entries falling on the diagonal than there were in Tables 3.10 and 3.11, indicating significantly better agreement between the two clustering methods. This stronger agreement is reflected in the agreement measures reported in Table 3.9.

The gold standard versus k-means comparison was arbitrarily chosen to illustrate the underlying frequency tables used for the cluster comparisons. Such tables exist for each cluster comparison reported in this section. The best case cluster labeling algorithm resulted in the most entries falling on the diagonal and thus indicates the best cluster agreement. The naïve and ranked based cluster labeling algorithms had fewer entries falling on the diagonal, and indicate a weaker cluster agreement than actually exists.

Table 3.13 repeats the above analysis using the Pearson correlation similarity measure. The Pearson correlation similarity measure is frequently employed when performing cluster analysis using microarray data and the dissimilarity is calculated as (1 – the usual Pearson correlation coefficient).

Table 3.13: Cluster Agreement for the Pearson Correlation Similarity Measure

Comparison	**Kappa**			**Weighted Kappa**			**RI**	**ARI**
	N	**R**	**BC**	**N**	**R**	**BC**		
GS vs. AL	0.24 (0.12,0.36)	0.21 (0.08,0.34)	0.35 (0.22,0.48)	0.35 (0.18,0.53)	0.08 (0*,0.24)	0.41 (0.24,0.59)	0.84	0.37
GS vs. KM	0* (0*,0.01)	0.04 (0*,0.15)	0.50 (0.36,0.63)	0.13 (0*,0.29)	0.06 (0*,0.20)	0.38 (0.19,0.57)	0.84	0.34
AL vs. KM	0.20 (0.08,0.31)	0* (0*,0.08)	0.26 (0.14,0.37)	0.62 (0.52,0.72)	0* (0*,0.04)	0.47 (0.35,0.60)	0.83	0.43

where RI = Rand Index, ARI = Adjusted Rand Index
GS = gold standard, KM = k-means, AL = average linkage
N = naïve labeling, R = ranked labeling, BC = best case labeling
Numbers in ()'s represent 95% confidence intervals
* = negative calculated value was set to zero

The results shown in Table 3.13 are discussed below. Note that the gold standard, or "true" clustering, remains the same regardless of the distance measure employed. Once again, for the kappa statistic all three comparisons show better agreement (0.35, 0.50, 0.26) for the best case cluster labeling algorithm than for the ranked and naïve algorithms. The weighted kappa statistics vary widely and a general statement cannot be made about them. However, the maximum weighted kappa statistic (0.62) occurs for the naïve cluster labeling algorithm for the average linkage versus k-means comparison. The adjusted Rand index is also a maximum (ARI = 0.43) for this comparison, and the Rand index (0.83) is very close to its maximum value of 0.84 for this comparison. Interpreting the adjusted Rand index statistic, the average linkage and k-means clustering methods agree quite strongly (ARI = 0.43) followed by the gold standard versus average linkage comparison (ARI = 0.37) and the gold standard versus k-means comparison (ARI = 0.34). Finally, by comparing Tables 3.9 and 3.13, it can be seen that the Pearson correlation similarity measure did a better job than the Euclidean distance measure at reproducing the true clusters using the average linkage and k-means clustering algorithms. The gold standard versus k-means comparison is identical for both the Euclidean distance and Pearson correlation similarity measures. This indicates that the k-means clustering algorithm is less sensitive to the choice of distance measure for these data than the average linkage clustering algorithm.

The comparison of the Euclidean distance and Pearson correlation similarity measures is formally made in Table 3.14. Cross comparisons are made between the average linkage and k-means clustering algorithms with the Euclidean distance and Pearson correlation similarity measures.

Table 3.14: Cluster Agreement for the Euclidean Distance versus the Pearson Correlation Similarity Measures

Comparison	**Kappa**			**Weighted Kappa**			**RI**	**ARI**
	N	**R**	**BC**	**N**	**R**	**BC**		
ALE vs. ALP	0* (0*,0.04)	0* (0*,0.01)	0.09 (0.00,0.17)	0* (0*,0.01)	0* (0*,0.03)	0* (0*,0.04)	0.45	0*
KME vs. KMP	1.0 (1.0,1.0)	1.0 (1.0,1.0)	1.0 (1.0,1.0)	1.0 (1.0,1.0)	1.0 (1.0,1.0)	1.0 (1.0,1.0)	1.0	1.0
ALE vs. KME/KMP	0* (0*,0.07)	0.07 (0.01,0.12)	0.25 (0.13,0.38)	0* (0*,0.01)	0.39 (0.25,0.53)	0.11 (0*,0.25)	0.59	0.24
KME vs. ALP	0.20 (0.08,0.31)	0* (0*,0.08)	0.36 (0.24,0.49)	0.62 (0.52,0.72)	0* (0*,0.04)	0.54 (0.39,0.69)	0.83	0.43

where RI = Rand Index, ARI = Adjusted Rand Index
ALE and ALP = average linkage with Euclidean distance and Pearson correlation similarity measures
KME and KMP = k-means clustering with Euclidean distance and Pearson correlation similarity measures
N = naïve labeling, R = ranked labeling, BC = best case labeling
Numbers in ()'s represent 95% confidence intervals
* = negative calculated value was set to zero

The average linkage clustering results are very different when using the Euclidean distance measure versus the Pearson correlation similarity measure. All but one of the agreement measures in row one of Table 3.14 is zero which indicates poor (chance) agreement for the average linkage clustering method using the Euclidean distance versus the Pearson correlation similarity measures. In contrast, all of the agreement measures in row 2 of Table 3.14 are one. This indicates that the k-means clustering algorithm agrees perfectly when using the Euclidean distance versus the Pearson correlation similarity measures. The k-means clustering algorithm is not as dependent as the average linkage clustering algorithm on the choice of a distance measure for this data set. The average linkage clustering algorithm using the Euclidean distance measure is compared to the k-means algorithm using both the Euclidean distance and Pearson correlation similarity measures. The average linkage and Euclidean distance measure combination agrees weakly (RI = 0.59, ARI = 0.24) with the k-means algorithm results. Finally, the k-means algorithm using the Euclidean distance measure compares favorably with the average linkage algorithm using the Pearson correlation similarity measure (RI = 0.83, ARI = 0.43).

3.5 Comparison of Independent Microarray Experiments

Section 3.4 examined various cluster agreement measures and presented a microarray example using data from Ross *et al.* (2000). As mentioned in Section 3.2, this same experiment was independently replicated twice using Affymetrix microarray designs. While these samples have many of the same genes in common, there is insufficient information on gene labels in the publicly available data files to match up individual genes between the three data sets. However, the 60 tumor cell lines are readily matched. This section compares the cell line clustering results of the average linkage and k-means clustering methods with the gold standard using the Pearson correlation similarity measure for the two Affymetrix data sets. Finally, the cluster results for the three data sets are compared with each other. The cluster results from the three data sets should ideally be very similar to each other. Note that no filtering is done in these analyses. Filtering can help to reduce noise and to remove outlying observations which might bias the cluster results. Chapter 2 contains details on filtering methods, as filtering is an important part of a complete microarray data analysis.

The Millenium pharmaceutical company data set (discussed in Section 3.2) contains gene expression data on 5,611 genes from 60 cancer cell lines. These 60 cell lines should segregate into 9 known tumor types. The actual tumor groups are called the gold standard. Table 3.15 shows how well the various clustering methods agree using the Pearson correlation similarity measure.

Table 3.15: Millenium Data Cluster Agreement for the Pearson Correlation Similarity Measure

Comparison	**Kappa**			**Weighted Kappa**			**RI**	**ARI**
	N	**R**	**BC**	**N**	**R**	**BC**		
GS vs. AL	0* (0*,0.07)	0* (0*,0.07)	0.09 (0*,0.20)	0.06 (0*,0.15)	0* (0*,0.06)	0.06 (0*,0.16)	0.52	0*
GS vs. KM	0.04 (0*,0.14)	0.02 (0*,0.12)	0.15 (0.02,0.27)	0.11 (0*,0.29)	0.04 (0*,0.17)	0.12 (0*,0.32)	0.76	0*
AL vs. KM	0.11 (0.02,0.21)	0.03 (0*,0.12)	0.20 (0.08,0.32)	0.07 (0*,0.17)	0.15 (0*,0.31)	0.12 (0*,0.26)	0.65	0.26

where RI = Rand Index, ARI = Adjusted Rand Index
GS = gold standard, KM = k-means, AL = average linkage
N = naïve labeling, R = ranked labeling, BC = best case labeling
Numbers in ()'s represent 95% confidence intervals
* = negative calculated value was set to zero

Regardless of the cluster labeling algorithm employed, the kappa and weighted kappa statistics in Table 3.15 are poor. The maximum kappa is 0.20, which indicates weak agreement. Notice that although the average linkage and k-means cluster algorithms agree somewhat with each other (ARI = 0.26), both algorithms reproduce the gold standard clusters at chance levels. As seen by comparing Tables 3.13 and 3.15, the Ross *et al.* (2000) data clustering results reproduce the actual clusters much better than the Millenium pharmaceutical data. This may be due in part to having fewer genes in the Millenium experiment or more noise present due to not filtering the data. Perhaps some of the genes that are present in the Ross experiment but absent in the Millenium experiment are very informative with regards to clustering the tumor cell lines.

The MIT data set (discussed in Section 3.2) contains gene expression data on 7,129 genes from 60 cancer cell lines. As before, these 60 cell lines should segregate into 9 known tumor types. Table 3.16 shows how well the various clustering methods agree using the Pearson correlation similarity measure.

Table 3.16: MIT Data Cluster Agreement for the Pearson Correlation Similarity Measure

Comparison	**Kappa**			**Weighted Kappa**			**RI**	**ARI**
	N	**R**	**BC**	**N**	**R**	**BC**		
GS vs. AL	0.01 (0*,0.08)	0.03 (0*,0.11)	0.15 (0.04,0.26)	0.04 (0*,0.15)	0.07 (0*,0.19)	0.11 (0.01,0.21)	0.49	0.02
GS vs. KM	0* (0*,0.01)	0* (0*,0.01)	0.34 (0.20,0.48)	0.06 (0*,0.22)	0.02 (0*,0.18)	0.41 (0.24,0.58)	0.82	0.16
AL vs. KM	0.04 (0*,0.11)	0.03 (0*,0.09)	0.14 (0.04,0.24)	0.05 (0*,0.14)	0.12 (0*,0.26)	0.09 (0*,0.21)	0.55	0.13

where RI = Rand Index, ARI = Adjusted Rand Index
GS = gold standard, KM = k-means, AL = average linkage
N = naïve labeling, R = ranked labeling, BC = best case labeling
Numbers in ()'s represent 95% confidence intervals
* = negative calculated value was set to zero

For the MIT data in Table 3.16, the maximum best case clustering labeling kappa and weighted kappa statistics indicate that the gold standard and k-means clusters agree the best (kappa = 0.34, adjusted kappa = 0.41). The Rand and adjusted Rand indices further support this conclusion (RI = 0.82, ARI = 0.16). The average linkage and k-means cluster algorithms agree weakly with each other (RI = 0.55, ARI = 0.13). By comparing Tables

3.14, 3.15, and 3.16, it can be seen that the Ross data set reproduces the actual tumor clusters best, with the MIT data set second best and the Millenium group last. There is a possibility that since the Millenium data was collected mainly for technology demonstration purposes, the data was not subjected to a high level of quality control and, as a consequence, the clustering algorithms do not reproduce the actual clusters as well.

The final comparison examines the cluster agreement between the Ross, Millenium, and MIT data sets introduced in Section 3.2. These comparisons are presented in Table 3.17.

Table 3.17: Cluster Agreement of 3 Data Sets Average Linkage and K-means Cluster Algorithms Using Pearson Correlation Similarity Measures

Comparison	**Kappa**			**Weighted Kappa**			**RI**	**ARI**
	N	**R**	**BC**	**N**	**R**	**BC**		
AL(R)/AL(Ml)	0.06 (0*,0.17)	0* (0*,0.03)	0.01 (0*,0.13)	0.15 (0.02,0.27)	0* (0*,0.01)	0* (0*,0.14)	0.51	0*
AL(R)/AL(MIT)	0.03 (0*,0.10)	0* (0*,0.02)	0.07 (0*,0.15)	0.00 (0*,0.14)	0* (0*,0.09)	0.09 (0*,0.27)	0.48	0.01
AL(Ml)/AL(MIT)	0* (0*,0.03)	0.01 (0*,0.16)	0.04 (0*,0.22)	0* (0*,0.01)	0.08 (0*,0.35)	0* (0*,0.08)	0.49	0*
KM(R)/KM(Ml)	0* (0*,0.04)	0.02 (0*,0.13)	0.10 (0*,0.23)	0* (0*,0.12)	0.06 (0*,0.24)	0.06 (0*,0.24)	0.71	0*
KM(R)/KM(MIT)	0.05 (0*,0.14)	0* (0*,0.09)	0.16 (0.05,0.28)	0* (0*,0.07)	0.14 (0*,0.31)	0.10 (0*,0.25)	0.76	0.08
KM(Ml)/KM(MIT)	0* (0*,0.01)	0* (0*,0.06)	0.20 (0.08,0.31)	0* (0*,0.01)	0* (0*,0.08)	0.08 (0*,0.22)	0.76	0.07

where RI = Rand Index, ARI = Adjusted Rand Index
KM = k-means, AL = average linkage
R = Ross data set, Ml = Millenium Data Set, MIT = MIT data set
N = naïve labeling, R = ranked labeling, BC = best case labeling
Numbers in ()'s represent 95% confidence intervals
* = negative calculated value was set to zero

All of the comparisons in Table 3.17 are made using the Pearson correlation similarity measures. Examining the average linkage results shows that for the best case cluster labeling, the kappa (0.07) and weighted kappa (0.09) statistics take on their maximum values for the Ross versus MIT data set comparisons. These values indicate only weak agreement. The Rand index is highest for the Ross versus Millenium data set comparisons (RI = 0.51). However, this index differs only slightly from the Rand index values for the other two comparisons. Finally, the only non-zero adjusted Rand index is for the Ross versus MIT data set comparison (RI = 0.01). These results indicate that the cluster agreement is strongest between the Ross and MIT data sets.

For the k-means cluster comparisons in Table 3.17, the kappa statistic is greatest (0.20) for the Millenium versus MIT data set comparison. However, the weighted kappa statistic is greatest (0.10) for the Ross versus MIT data set comparison. The Rand index is tied at a maximum value for the Ross versus MIT data set comparison (RI = 0.76), while the adjusted Rand index achieves its maximum value for this comparison (ARI = 0.08). Notice that the k-means cluster result comparisons tend to have greater Rand and adjusted Rand index values than the average linkage cluster result comparisons. This suggests that the k-means algorithm may be doing a better job at extracting cluster information than the average linkage algorithm. Table 3.17 illustrates that even among microarray experiments that target the same genes and cell lines, there is not good agreement on how to cluster the cell lines into the nine tumor types.

3.6 Conclusion

This chapter presented several methods for labeling clusters and measuring the similarity between the cluster results. This is meant to be an introduction to these issues, as there is little in the literature about labeling clusters prior to comparison. The comparisons were made using cluster results from the average linkage and k-means clustering algorithms using the Euclidean distance and Pearson correlation similarity measures. Three microarray data sets were examined. Empirical evidence suggested that the choice of the cluster agreement measure and how one labels the clusters strongly effects how well the clusters are said to agree. The best case cluster labeling algorithm consistently improved the agreement between two competing clustering algorithms. One could use this algorithm to compare three or more clustering techniques by running the algorithm for each pairwise comparison of the methods. This labeling algorithm may not have arrived at the optimum cluster labels, but has the advantage of being easy to automate. A more exhaustive manual assignment of cluster labels may further improve cluster agreement, but this is not practical for large data sets. The measures of agreement should be treated relative to each other rather than strictly interpreted based on magnitudes.

The results of the average linkage clustering algorithm appeared to be more sensitive to a change in distance measures than the k-means clustering algorithm. Personal experience in analyzing a range of microarray data suggests that the Pearson correlation similarity measure helps to replicate the true clusters the best. The Pearson correlation is defined as:

$$\rho_{i,j} = \frac{\sum_{i=1}^{G}\left(g_{i1}-\overline{g}_{.1}\right)\left(g_{i2}-\overline{g}_{.2}\right)}{\sqrt{\sum_{i=1}^{G}\left(g_{i1}-\overline{g}_{.1}\right)^{2}}\sqrt{\sum_{i=1}^{G}\left(g_{i2}-\overline{g}_{.2}\right)^{2}}} \qquad (3.13)$$

where the *i* and *j* subscripts refer to specific genes and *G* is the total number of genes. However, it is not possible to make a general statement as to what distance measure to use, as it varies according to the data.

The k-means algorithm clusters the data into a user specified number of clusters. An advantage of the average linkage clustering algorithm is that the data can be represented hierarchically in a dendrogram which the user can analyze visually. Average linkage clustering does not require a rigid adherence to a given number of clusters. Dendrograms are interpreted at a specific level according to the distance between the elements in clusters. One can think of a dendrogram as a tree in which the longer branches indicate clusters which are less homogenous. For the average linkage cluster results presented in this chapter, the dendrogram was interpreted at the distance for which the tree contained nine clusters because there are nine known tumor groups. This may not be the optimal number of clusters, as the nine tumor groups may be further divisible based on their gene expression patterns. For most applications, the true number of groups is unknown.

The results of the cluster analysis of the three data sets (Ross, MIT, and Millenium) did not compare favorably. This illustrates the difficulty in comparing microarray data generated from different experiments. This disagreement could have come from attempting to compare experiments run for different purposes using the fundamentally different cDNA and Affymetrix designs and perhaps would have improved had a filtering method been applied. (Data filtering is discussed in Chapter 2.) The differences also could have arisen from using independent cell samples or varying experimental conditions and procedures. One can imagine how these difficulties might be compounded when clustering over thousands of genes instead of 60 cancer cell lines. Solving these problems is a focus of microarray research today.

The average linkage and k-means clustering methods are both non-parametric and require the user to select a desired number of clusters (either by choosing a level to cut the dendrogram at for the average linkage case or choosing a number of clusters before running the model for the k-means case). Choosing the appropriate number of clusters is difficult since there is no statistic available indicating which clustering results in the "best" groups of observations. One common approach to this problem is to try to select a method which minimizes the within cluster variances and maximizes the between cluster variances. Another possibility is to apply a parametric clustering technique, which requires making certain assumptions about the distribution of the data. The advantage of using a parametric clustering technique is that statistically based criteria are available for evaluating how well the model clusters the data. Parametric clustering techniques are discussed in Chapters 4 and 5.

Chapter 4

Parametric Clustering

4.1 Introduction

4.1.1 Parametric Models for Clustering

Parametric models are increasingly used in the field of cluster analysis (McLachlan and Basford, 1988; Kerr and Churchill, 2001). As Aitkin and Aitkin (1996) wrote, "When clustering samples from a population, no clustering method is *a priori* believable without a statistical model." Parametric models help to formalize cluster analysis by allowing statistical models to be developed and tested. One type of parametric cluster analysis uses a class of models known as mixture models. Using mixture models, a likelihood based approach is applied to cluster experimental data. The mixture models approach assumes that the observations for the entities to be clustered are from a mixture of a specified number of groups in various proportions. By assuming a parametric form for the density function in each group, a likelihood function can be formed in terms of a mixture density. The unknown parameters of the distribution can be estimated by the method of maximum likelihood. The maximum likelihood (ML) equations for mixture models are nonlinear, and, therefore, they are solved iteratively. This process leads to estimates of cluster specific parameters as well as the proportion of observations falling in each cluster and the posterior probability of each observation falling in a specific cluster. Clustering proceeds by assigning each entity to a group based on the relative value of the estimated posterior probability of belonging to that group compared with the posterior probabilities of belonging to the other groups. Karl Pearson (1894) was one of the first to apply parametric mixture models. Pearson fitted a two component univariate normal mixture model using the method of moments to a set of measurements on the ratio of forehead to body length on 1,000 crabs. This data set is reanalyzed in Section 4.3.6.

The type of mixture model used in cluster analysis is often called a finite mixture model because of the assumption that there are a finite number, C, of groups in the data. If at least one of the mixture components comes from a discrete distribution (such as the multinomial), the mixture model approach is sometimes called latent class analysis (McLachlan and Peel, 2000; Everitt *et al.*, 2001). Latent class analysis dates back to Lazarsfeld (1950) and has been widely applied in the social sciences. This dissertation focuses on finite mixture models.

One of the first applications of finite mixture models to microarray data was by Ghosh and Chinnaiyan (2002). They define normal mixture models and develop the Expectation/Maximization algorithm for fitting these models using the approach discussed in Section 4.3. They apply a variety of constraints on the cluster structure. The method proposed in this dissertation relaxes some of these constraints. One of the advantages for using mixture models for microarray data is that they provide a statistical criterion for assessing the number of clusters present in the data. A strong assumption made in fitting mixture models to microarray data is that the genes are independent and identically distributed according to the mixture density defined in Equation 4.2. More work is needed in order to relax this assumption. (One way of doing this may be to fit multivariate mixture models, as suggested in Chapter 6.) Ghosh and Chinnaiyan (2002) suggest the Bayesian Information Criterion (discussed in Section 4.3.5) as a statistic useful for comparing mixture models having different numbers of clusters. They apply average linkage hierarchical clustering using the Euclidean distance measure to obtain starting values for fitting mixture models. They comment that convergence problems frequently occurred when using these starting values and suggest using random partitions of the data to generate starting values in such cases. The results of a k-means cluster analysis to provide the starting values is found in this dissertation to lead to better convergence properties. When clustering across samples, Ghosh and Chinnaiyan (2002) comment that the number of samples is typically much smaller than the number of genes. For such situations, they suggest reducing the dimension of the data by using principal components analysis. This method does not reduce the dimensionality enough in larger microarray experiments. An alternative method of filtering the genes is proposed in Section 4.3.8. Ghosh and Chinnaiyan (2002) analyze two microarray datasets. The first is from a malignant melanoma study reported by Bittner *et al.* (2000). There were 31 melanoma samples and 3,613 genes included in the analysis. The data were normalized using the usual log corrected intensity ratios. They found two clusters of melanomas – one of size 19 and the other of size 12. They did not report the results of clustering the genes. The second microarray dataset was from a prostate cancer study (Dhanasekaran *et al.*, 2001). There were 26 samples from 5 different biopsy locations. There were 3,955 genes included in the analysis. They clustered the genes and investigated the results for cluster sizes of 250, 1000, and 2000. For 250 clusters, several biologically plausible groups of genes were found.

Several other papers which apply mixture models to microarray data have been published. Mjolsness *et al.* (2000) apply mixture models to determine the number of clusters present in a microarray data set. They select the "optimal" model based on the maximum log likelihood. They discuss a process known as circuit inference which uses simulated annealing to establish a network of connection strengths similar those of a neural network (neural networks are briefly discussed in Chapter 6). McLachlan *et al.* (2002) suggest applying mixture models as a gene filtering tool. They select a subset of genes by fitting mixtures of *t* distributions to rank the genes in order of increasing size of the likelihood ratio statistic for the test of including the gene in the model versus not. They introduce a software package called EMMIX-GENE to perform this ranking. Pan (2002) applies the normal mixture model suggested by Ghosh and Chinnaiyan (2002) to cluster microarray data on the susceptibility of rats to ear infections. He found three clusters, two of which were attributed genes with no altered expression levels and one of which contained at least 30 genes having differential expression levels. Allison *et al.* (2002) apply mixture models to cluster p-values indicating whether or not differential gene expression is present. This approach is only relevant in the sense that mixture models were applied.

4.1.2 Introduction to the Expectation/Maximization (EM) Algorithm

The Expectation/Maximization (EM) algorithm is a two-stage iterative algorithm useful for calculating likelihood estimates when the data are incomplete. Incomplete data occur when observations on random variables are missing or censored. In the mixture model case, the indicator variables that assign elements to the various clusters are unobserved and can be treated as missing, thus defining the incomplete data. The EM algorithm allows parameter estimates to be obtained under such circumstances. The EM algorithm consists of expectation and maximization steps. The expectation step estimates the incomplete data by calculating expected values conditional on the observed data. Once the incomplete data are estimated in the expectation step, the maximum likelihood estimates of the parameters are calculated in the maximization step. The EM algorithm requires starting values for the parameter estimates to be input for the first expectation step. The EM algorithm is developed for the mixture model case in Section 4.3.2. In general, the EM algorithm consists of the following steps:

1. Replace the missing values by conditional expectations
2. Estimate the parameter values by the maximum likelihood method.
3. Iterate
    a. Step 1 using the estimated parameter values as the true values.
    b. Step 2 using the estimated values as "observed" values, iterating until convergence.

The ideas underlying the EM algorithm were first discussed by Orchard and Woodbury (1972). The EM algorithm was formally developed by Dempster *et al.* (1977). The application of the EM algorithm to clustering problems is discussed in this dissertation. For more detailed information on the development and use of the EM algorithm, see Little and Rubin (1987) or McLachlan and Krishnan (1997).

4.1.3 Microarray Data

Microarray experiments are becoming cheaper and easier to perform with the help of popular designs such as the dye-swap, reference, and loop designs (Kerr and Churchill, 2001). The two major platforms on which microarray experiments are performed are the Affymetrix and complementary DNA (cDNA). The focus of this chapter is on the analysis of cDNA arrays. Regardless of the platform chosen, microarrays are setup as two dimensional plates (often made of glass) containing a large number of wells or areas onto which the probes are placed. Through a highly automated process, the probes are placed, or spotted, onto the microarrays. The identity of the spot is retained by keeping track of its position on the array. Standard identifying characteristics include, for example, the cell type or variety, replicate number, the gene which the probe represents, etc. Each probe is typically spotted multiple times on an array in order to help to control technical variability. For more information on microarray designs, see Section 1.1.

For cDNA microarrays, mRNA is extracted from the cells and hybridized to form the cDNA samples. These samples are usually labeled with a red (Cy3) and/or a green (Cy5) dye. The microarrays are "read" by shining a laser through a particular spot on the array and recording the fluorescence value. The fluorescence value is indicative of the abundance of the gene for that sample. Normalization methods attempt to transform the data from different chips in such a way that they are comparable. Normalization is often performed using some function involving the background values in order to more accurately compare arrays having different backgrounds. One typical normalization approach is to subtract the background fluorescence value from the fluorescence value for the signal of interest. A transformation, such as the logarithm, is usually performed on this corrected value. The use of this background corrected value helps to separate the signal from the background. For more information on microarray data normalization, see Quackenbush (2001, 2002). Typical analyses performed on the fluorescence values include cluster and classification analyses which may attempt to find genes which are expressed in specific biological processes.

4.1.4 Chapter Contents

This dissertation applies parametric clustering models to both simulated and experimental microarray data. The advantages of using parametric models for clustering are discussed. Mixture model theory is introduced. The Expectation/Maximization (EM) algorithm is outlined as an iterative likelihood technique useful for obtaining parameter estimates for the mixture model. Microarray data are described. Prior applications of mixture models to microarray data are reviewed. The data set from Ross *et al.* (2000) is introduced. The one dimensional normal mixture model is formulated. The theory for implementing the EM algorithm is developed. An extension of the EM algorithm called the EM/Newton-Raphson hybrid algorithm is explored for accelerating convergence. Confidence interval formulas are derived for the estimates of the parameters and the posterior probabilities of a given observation belonging to a specific cluster. Techniques for selecting the model having the optimal number of clusters are described. 2-DCluster, a software package developed for this dissertation, implements these methods and calculates the appropriate confidence intervals (see the 2-DCluster Appendix for the software documentation and availability). Data are simulated, analyzed, and discussed. Experimental data from the Ross *et al.* (2000) microarray experiment are analyzed, followed by a discussion of the results.

4.2 Introduction to the Ross *et al.* (2000) Data Set

The National Cancer Institute's (NCI) Developmental Therapeutics Program (DTP) has intensively studied 60 cancer cell lines (Ross *et al.*, 2000), which are known as the NCI 60. The 60 cell lines are from the tumor types listed in Table 4.1.

Table 4.1: Nine Tumor Types in the Ross Data Set

Tumor Type	Number of Cell Lines
Breast Cancer	8
Central Nervous System Cancer	6
Colon Cancer	7
Leukemia	6
Melanoma	8
Non-Small Cell Lung Cancer	9
Ovarian Cancer	6
Prostate Cancer	2
Renal Cancer	8

Using the two color Complementary DNA (cDNA) design, microarrays were prepared by robotically spotting 9,703 human cDNAs on glass microscope slides. The cDNAs included approximately 8,000 unique genes. Each hybridization compared Cy5 labeled cDNA reverse transcribed from mRNA isolated from one of the cell lines with Cy3 labeled cDNA reverse transcribed from a reference mRNA sample. The reference sample, used in all of the hybridizations, was prepared by combining an equal mixture of mRNA from 12 of the cell lines. Only 6,165 genes had complete data for all of the 60 cell lines. These values were transformed using the usual log background corrected ratio for the two channels. The investigators presented the results from an average linkage cluster analysis using Pearson's correlation as the similarity measure. They found that cell lines with common tissues of origin tended to cluster together. Cluster analyses were repeated using different subsets of genes to assess cluster robustness. The authors conclude that the clusters seem to be reasonably robust. A major goal of this experiment was to examine the chemosensitivity of the NCI 60 to about 70,000 different chemical compounds. The chemosensitivity data has been analyzed by Ross *et al.* (2000) as well as in separate studies by Paull *et al.* (1989), van Osdol *et al.* (1994), and Weinstein *et al.* (1992, 1997). The chemosensitivity data is not discussed in this dissertation.

4.3 The One Dimensional Normal Mixture Model

Mixture models may be formed from a variety of component distributions. This dissertation focuses on a finite normal mixture model in which the mixture is formed from *C* univariate normal densities in various proportions (π_C). (For more information on mixture densities having non-normal components or multivariate component densities, see Everitt and Hand (1981) or Titterington *et al.* (1985)).

Normal mixture distributions have three general forms. The first looks like a normal density curve as shown in Figure 4.1. This mixture occurs when the normal components have similar means but different variances. This type of mixture is the most difficult to fit since an identifiability problem will likely arise.

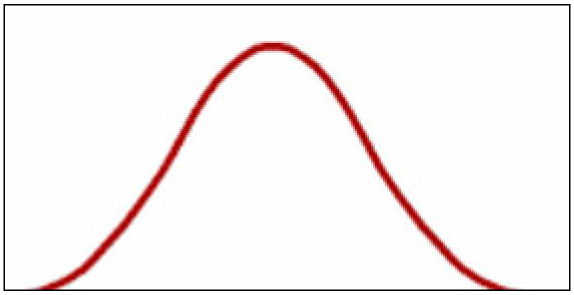


Figure 4.1: Normal Mixture with Similar Means but Different Variances

The second type of normal mixture density takes on a multimodal form. Figure 4.2 shows such a density for a two component normal mixture case. This mixture occurs when the normal components have different means which are "far enough" apart in relation to their variances to form distinct modes on the curve. The variances could also differ for this type of normal mixture. Mixture models work best for estimating the normal component parameters of such distributions.

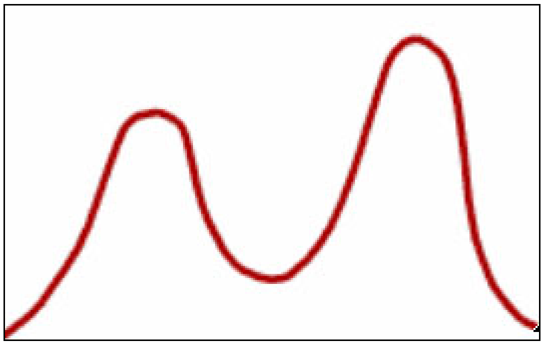


Figure 4.2: Two Component Normal Mixture

The third type of normal mixture density takes on various shapes and usually occurs when the means are similar but the variances differ widely. Figure 4.3 shows an example of a normal mixture for the skewed case. Similarly to unimodal mixtures (Figure 4.1), it is difficult to accurately estimate the parameters for these types of mixtures.

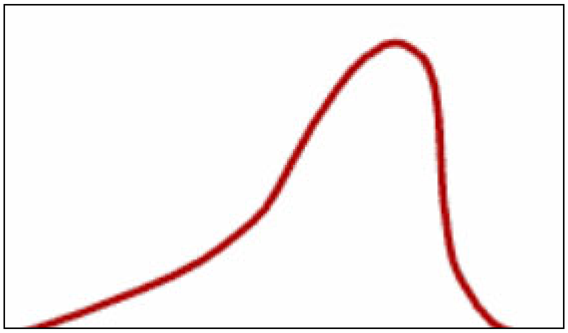


Figure 4.3: Skewed Normal Mixture

In this dissertation, the applications considered are expected to be multimodal (Figure 4.2). The parameters could become increasingly difficult to estimate and models could take longer to converge the further a mixture distribution deviates from this form. For the case of microarray data, there is no strong reason to suspect that only the variances of the gene expression values differ significantly but not the means. The multimodal nature of the data is not expected to be as dramatic as that shown in Figure 4.2 when plotted. However, the data are anticipated to be "close enough" to this form to allow the parameters to be estimated. Additional assumptions are that there are a small number of clusters and that not all of the observations come from different distributions.

4.3.1 Introduction to the Model

The normal mixture distribution is derived from a mixture of C normal distributions $\phi\left(x;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)$, where μ_k is the mean for the k^{th} component and σ_k^2 is the variance for the k^{th} component, $k = 1,\ldots,C$. The density $\phi\left(x;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)$ is standard notation for the normal density,

$$\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left[-\frac{1}{2\sigma_k^2}\left(x-\mu_k\right)^2\right]. \tag{4.1}$$

A random variable, Y, is distributed as a normal mixture distribution if the density of Y is of the form:

$$f_{y_i}\left(y_i;\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right) = \sum_{k=1}^{C} \pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right), \tag{4.2}$$

where π_k is the proportion of observations falling in the k^{th} group, $\sum_{k=1}^{C} \pi_k = 1$, $0 < \pi_k < 1$, and $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\sigma}^2$ are vectors containing $(\pi_1, \ldots, \pi_C)$, $(\mu_1, \ldots, \mu_C)$, and $\left(\sigma_1^2, \ldots, \sigma_C^2\right)$. This restriction implies that only C-1 of the proportions need to be estimated, since $\pi_C = 1 - \sum_{k=1}^{C-1} \pi_k$.

The distributional properties of *Y* are described below for the two component mixture case. Suppose that the first component of the mixture density has the distribution $\phi_{x_1}\left(x;\ \mu_1,\ \sigma_1^2\right)$ and that the second component has distribution $\phi_{x_2}\left(x;\ \mu_2,\ \sigma_2^2\right)$. A random variable, *Y*, is distributed as a two component normal mixture distribution having a density of the form:

$$f_{y_i}\left(y_i;\ \pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\right) = \pi\phi\left(y_i;\ \mu_1,\ \sigma_1^2\right) + (1-\pi)\phi\left(y_i;\ \mu_2,\ \sigma_2^2\right). \tag{4.3}$$

The expected value of *Y* is found by

$$E[Y] = \int_{-\infty}^{\infty} y_i f_{y_i}\left(y_i;\ \pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2\right)\, \partial y_i. \tag{4.4}$$

Using the usual rules for calculating expected values,

$$\begin{aligned} E\left[Y\right] &= \int_{-\infty}^{\infty} y_i\left[\pi\phi\left(y_i;\ \mu_1,\ \sigma_1^2\right)+\left(1-\pi\right)\phi\left(y_i;\ \mu_2,\ \sigma_2^2\right)\right]\partial y_i \\ &= \pi\int_{-\infty}^{\infty} y_i\phi\left(y_i;\ \mu_1,\ \sigma_1^2\right)\partial y_i+\left(1-\pi\right)\int_{-\infty}^{\infty} y_i\phi\left(y_i;\ \mu_2,\ \sigma_2^2\right)\partial y_i \\ &= \pi\mu_1+\left(1-\pi\right)\mu_2. \end{aligned} \tag{4.5}$$

The general form for the expected value of *Y* in a *C* component mixture can be shown to be

$$E\left[Y\right]=\sum_{k=1}^{C}\pi_k\mu_k. \tag{4.6}$$

The variance of *Y* by definition is $V\left[Y\right]=E\left[Y^2\right]-\left(E\left[Y\right]\right)^2$. The expected value of *Y* is already calculated (Equation 4.5). The expected value of Y^2 is

$$E\left[Y^2\right]=\int_{-\infty}^{\infty} y_i^2 f_{y_i}\left(y_i;\ \pi,\mu_1,\mu_2,\sigma_1^2,\sigma_2^2\right)\partial y_i. \tag{4.7}$$

Simplifying as in Equation 4.5 now for Y^2 and applying $E\left(Y^2\right)=\mu^2+\sigma^2$ for the $N\left(\mu,\sigma^2\right)$ case yields:

$$E\left[Y^2\right]=\pi\left(\mu_1^2+\sigma_1^2\right)+\left(1-\pi\right)\left(\mu_2^2+\sigma_2^2\right). \tag{4.8}$$

Plugging $E\left[Y\right]$ and $E\left[Y^2\right]$ into the variance equation yields:

$$V\left[Y\right]=\left[\pi\left(\mu_1^2+\sigma_1^2\right)+\left(1-\pi\right)\left(\mu_2^2+\sigma_2^2\right)\right]-\left[\pi\mu_1+\left(1-\pi\right)\mu_2\right]^2. \tag{4.9}$$

The general form for the variance of Y in a C component mixture can be shown to be

$$V\left[Y\right]=\sum_{k=1}^{C}\pi_k\left(\mu_k^2+\sigma_k^2\right)-\left[\sum_{k=1}^{C}\pi_k\mu_k\right]^2. \tag{4.10}$$

The shape of the two component normal mixture distribution depends on the proportion, component means, and component variances. Figure 4.2 gave an example of a two component mixture distribution. The following figures demonstrate what happens when the component means and variances change for a simulated two component normal mixture distribution.

Figure 4.4 shows the density for the two component normal mixture case with different means and equal variances. Note that two distinct modes are present which makes parameter estimation less difficult.

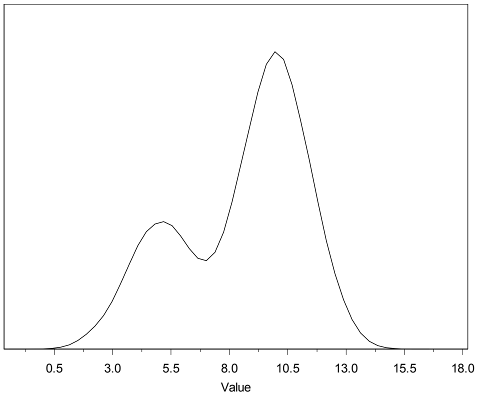


Figure 4.4: Two Component Normal Mixture Density
($N=10,000$, $\pi=0.3$, $\mu_1=5$, $\mu_2=10$, $\sigma_1^2=2$, $\sigma_2^2=2$)

Figure 4.5 shows the density for the two component normal mixture case with equal means and different variances.

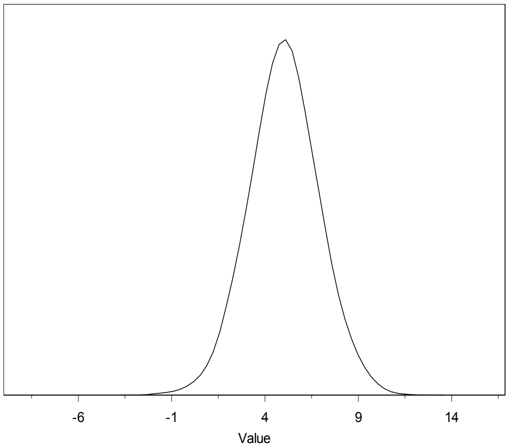


Figure 4.5: Two Component Normal Mixture Density
($N = 10,000$, $\pi = 0.3$, $\mu_1 = 5$, $\mu_2 = 5$, $\sigma_1^2 = 2$, $\sigma_2^2 = 4$)

It is very difficult to estimate parameters in such a situation because there are no distinct modes in the distribution due to the means being identical. Essentially, the mixture density is indistinguishable from a normal distribution or a *t* distribution. The curve is rather narrow and peaked due to the small variances.

Figure 4.6 shows the density for the two component normal mixture case with different means and different variances. As described earlier in this section, this is the type of mixture that is expected for the case of microarray data. This figure looks very similar to Figure 4.4, although the two modes are less distinct. The modes will become more apparent if the variances are small in relationship to their respective means. The parameters can be readily estimated in such a case.

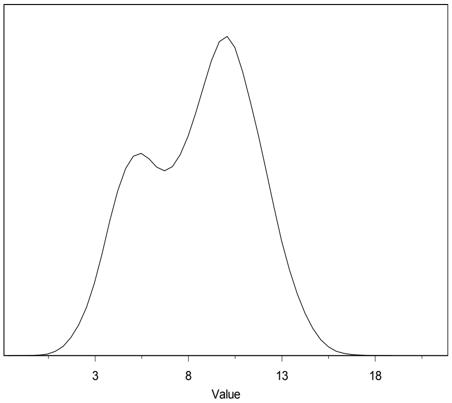


Figure 4.6: Two Component Normal Mixture Density ($N=10{,}000$, $\pi=0.3$, $\mu_1=5$, $\mu_2=10$, $\sigma_1^2=2$, $\sigma_2^2=4$)

If both of the means and both of the variances are the same, the mixture density is identical to the regular normal density. In this case, the data cannot be clustered, as only one group is present.

4.3.2 Parameter Estimation for the Normal Mixture Model

Suppose that a random sample of N observations is obtained from a normal mixture as defined in Section 4.3.1. The likelihood for the N observations is given by the joint density of the sample:

$$L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)=\prod_{i=1}^{N}\sum_{k=1}^{C}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right) \tag{4.11}$$

To obtain the maximum likelihood estimates (MLE’s), the log likelihood,

$$\ell\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)=\log\left[L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)\right]=\sum_{i=1}^{N}\log\left[\sum_{k=1}^{C}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)\right], \tag{4.12}$$

is maximized with respect to the parameters π_k, μ_k, and σ_k^2. The MLE's are found by taking the first partial derivatives of the log likelihood with respect to the parameters of interest, setting them equal to zero, and solving.

The first derivatives were given for the normal mixture distribution by Hasselblad (1966). For convenience, Hasselblad's notation is introduced below and is used from this point forward. Let *N* be the total number of observations with $i = 1, \ldots, N$. Let *C* be the number of clusters with $k = 1, \ldots, C$. Let y_i be the i^{th} observation, π_k be the proportion of total observations contained in cluster k, and μ_k and σ_k^2 be the mean and variance of the observations in cluster *k*, respectively. Let the distribution of the k^{th} normal mixture component be represented as:

$$f_{ik} = \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left[-\frac{1}{2\sigma_k^2}(y_i - \mu_k)^2\right]. \tag{4.13}$$

Note that the *i* subscript in Equation 4.13 is not strictly necessary, as the distribution only changes based on the cluster, not the observation. However, the *i* is included in order to be consistent with Hasselblad's notation. The normal mixture distribution of the random variable *Y* is written as:

$$F_i = f_{y_i}\left(y_i;\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma^2}\right) = \sum_{k=1}^{C} \pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right) = \sum_{\mathrm{k}=1}^{\mathrm{C}} \pi_{\mathrm{k}} f_{ik}. \tag{4.14}$$

The derivatives are calculated under the constraint that $\pi_C = 1 - \sum_{k=1}^{C-1} \pi_k$.

The first partial derivatives of the log likelihood are shown below.

$$\frac{\partial \ell}{\partial \pi_k} = \sum_{i=1}^{N} \frac{f_{ik} - f_{iC}}{F_i}, \ \ \mathrm{k} = 1, 2, \ldots, C - 1 \tag{4.15}$$

$$\frac{\partial \ell}{\partial \mu_k} = \sum_{i=1}^{N} \frac{\pi_k f_{ik} \left(y_i - \mu_k \right)}{\sigma_k^2 F_i}, \ \ \mathrm{k} = 1, 2, \ldots, C \tag{4.16}$$

$$\frac{\partial \ell}{\partial \sigma_k} = \sum_{i=1}^{N} \frac{\pi_k f_{ik}}{F_i} \left[\frac{\left(y_i - \mu_k \right)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right], \ \ \mathrm{k} = 1, 2, \ldots, C \tag{4.17}$$

These equations are nonlinear and therefore must be solved iteratively. The usual method for this circumstance is the Newton Raphson (NR) algorithm. However, the NR algorithm may have convergence problems and sometimes converges to a local maximum (Heath, 1997). Therefore, the Expectation/Maximization (EM) algorithm is suggested, as it should converge to the global maximum. The EM algorithm converges most of the time.

To develop the EM algorithm, the first requirement is the definition of the incomplete data (Dempster *et al.*, 1977). For the mixture model, the incomplete data are defined by the indicator variables that assign the observations to specific clusters. These indicator variables may be written as:

$$I_{ik} = \begin{cases} 1, \text{ if } y_i \in \text{ the } k^{th} \text{ cluster} \\ 0, \text{ otherwise} \end{cases}. \tag{4.18}$$

The distributions of these indicator variables are specified by the posterior probabilities, α_{ik}. In other words, $P(I_{ik} = 1 \mid y_i) = \alpha_{ik}$. In the Expectation stage of the EM algorithm, the complete data are obtained by estimating α_{ik}. The value α_{ik} represents the posterior probability of the i^{th} observation falling in the k^{th} cluster. The posterior probability is estimated by taking a weighted average over the C component densities. That is,

$$E[I_{ik} \mid \mathbf{y}] = P[I_{ik} = 1 \mid \mathbf{y}] = \hat{\alpha}_{ik} = \frac{\hat{\pi}_k \ \phi\left(y_i;\ \hat{\mu}_k,\ \hat{\sigma}_{\mathrm{k}}^2 \right)}{\sum_{r=1}^{C} \hat{\pi}_r \ \phi\left(y_i;\ \hat{\mu}_r,\ \hat{\sigma}_{\mathrm{r}}^2 \right)}. \tag{4.19}$$

Notice that the number of unknowns in this step is $3C - 1$. Thus, $3C - 1$ initial values need to be specified: C-1 for the proportions, C for the means, and C for the variances. After the initial iteration, new estimates for these values are obtained from the maximization step of the EM algorithm.

The maximization step of the EM algorithm uses the completed data to estimate the parameters of the distribution using maximum likelihood techniques. Once the observations falling into different groups are identified, obtaining the $3C - 1$ estimates of the proportions, means, and variances is straightforward and explicit solutions exist.

Equations 4.20 through 4.22 give the maximum likelihood estimates for the proportions, means, and variances for the normal mixture problem. For details of these derivations, see Appendix 4.1. The estimate for the proportion of observations falling in the k^{th} cluster, π_k, is:

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} \hat{\alpha}_{ik}}{N} \ . \tag{4.20}$$

The estimate for the mean of the k^{th} cluster, μ_k, is:

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} y_i \hat{\alpha}_{ik}}{\sum_{i=1}^{N} \hat{\alpha}_{ik}}. \tag{4.21}$$

The estimate for the variance of the k^{th} cluster, σ_k^2, is:

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\alpha}_{ik} \left(y_i - \hat{\mu}_k \right)^2}{\sum_{i=1}^{N} \hat{\alpha}_{ik}}. \tag{4.22}$$

For the case of homogenous variances among the C clusters, the σ_k^2's are first estimated for all of the clusters. Once these estimates are found, the new common variance is found by:

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{C} \hat{\sigma}_k^2 \left(\sum_{i=1}^{N} \hat{\alpha}_{ik} \right)}{N} \ . \tag{4.23}$$

Each iteration of the EM algorithm involves calculating equations 4.19 through 4.22 in sequence. For the first iteration, the starting values are used to calculate $\hat{\alpha}_{ik}$. At the end of each iteration, a check is made to see if the EM algorithm converged. There are a number of ways that this check can be performed. The approach taken by McLachlan and Peel (2000) is used in this dissertation. They check for convergence by examining all C of the proportion (π_k) estimates and seeing if the change from the previous iteration's estimate is less than some tolerance value. If any one of the C estimates has changed by some amount greater than the tolerance value, the algorithm continues. The justification for this stopping rule is that the proportions are the most important parameters to estimate accurately because they indicate the relative sizes of the clusters to the user and contain the $\sum_{i=1}^{N}\hat{\alpha}_{ik}$ terms which are involved in both the mean and variance estimates. Other stopping rules are possible, such as terminating when the means or variances change less than a specified tolerance between iterations. However, such rules would place too many restrictions on the algorithm and the convergence time could drastically increase.

Cluster membership is assigned by calculating the C estimates of the $\hat{\alpha}_{ik}$'s for all of the y_i's and assigning the observation to the cluster for which the posterior probability of belonging is the greatest. However, for values of $\hat{\alpha}_{ik}$ that are very close to each other, assigning an individual to a cluster based on the maximum posterior probability may not be optimal, since competing cluster assignments may be equally "good". All of the posterior probabilities are available from the mixture model fit, and the user can evaluate the effects of different cluster assignments. If two posterior probabilities are tied at the maximum value, one can either randomly select a cluster to assign the observation to or place the observation in both of the clusters. For posterior probabilities that are very close to each other, one could try different cluster assignments and evaluate how well the clusters fit the data using the statistics introduced in Section 4.3.5.

4.3.3 The EM/Newton-Raphson Hybrid Algorithm

The EM algorithm for mixture models, although an improvement over the Newton-Raphson algorithm, is often slow to converge. The convergence rate depends on several factors, such as the number of clusters, the number of observations, how close the parameters for the different clusters are to each other, which starting values are chosen, and so on. A number of approaches for speeding up the convergence of the EM algorithm in the mixture model case have been proposed in the literature (Aitkin and Aitkin, 1996; Neal and Hinton, 1998; Bradley *et al.*, 1999). This dissertation presents the hybrid method proposed by Aitkin and Aitkin (1996) that switches back and forth between the EM algorithm and the Newton-Raphson algorithm.

The EM algorithm always converges to the MLE (Dempster *et al.*, 1977) while the NR algorithm might converge to a local maximum or minimum or decrease the likelihood between iterations. Although the EM algorithm is much faster to converge to the right neighborhood, it is slow to reach the maximum. On the other hand, the NR algorithm is faster in reaching the maximum provided that it is in the right neighborhood. When the NR algorithm converges, its convergence rate is usually quadratic compared to linear for the EM algorithm (Aitkin and Aitkin, 1996). The EM algorithm traverses the likelihood surface in larger steps than the NR algorithm. Aitkin and Aitkin (1996) proposed the hybrid EM/NR algorithm to exploit the fast convergence of the EM algorithm with the local accuracy of the NR algorithm.

The EM algorithm is described for the normal mixture model case in previous sections. The Newton-Raphson algorithm makes use of a first order Taylor series expansion of the function being maximized (Heath, 1997):

$$f(x) \approx f(h) + f^{'}(h)(x-h). \tag{4.24}$$

This motivates the following iteration scheme, known as Newton's method:

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{'}(x_n)}, \tag{4.25}$$

where *n* represents the iteration number.

The Newton-Raphson method is readily extended to find the solutions to a set of simultaneous equations (Bickel and Doksum, 2001). The form of the new iterate may be expressed in matrix notation as:

$$\underline{\mathbf{X}}_{n+1} = \underline{\mathbf{X}}_n - \mathbf{H}\left(\underline{\mathbf{X}}_n\right)^{-1} \mathbf{B}\left(\underline{\mathbf{X}}_n\right), \tag{4.26}$$

where *n* is the iteration number, **X** is the vector of parameter estimates, **B** is the vector of first derivatives with respect to the parameters, and **H** is the Hessian matrix of second derivatives.

For example, in the case of a two component normal mixture, there are $3C-1=3*2-1=5$ parameters present. The 5x1 vector of parameters, **X**, takes the form:

$$\mathbf{X} = \begin{bmatrix} \pi \\ \mu_1 \\ \mu_2 \\ \sigma_1 \\ \sigma_2 \end{bmatrix}. \tag{4.27}$$

The 5x5 Hessian matrix, **H**, is composed of the second derivatives of the log likelihood with respect to the 5 parameters. Finally, the 5x1 vector, **B**, contains the first derivatives of the log likelihood with respect to the 5 parameters. The matrices **H** and **B** are shown in Equations 4.28 and 4.29, respectively. In general, a mixture model with C clusters has 3C – 1 parameters that must be estimated. The dimensions of **X**, **H**, and **B** increase in multiples of 3 as the number of clusters increases. The Newton-Raphson algorithm starts with the initial values and iterates Equation 4.26 until convergence is reached. Note that each iteration requires the inversion of a 3C - 1 dimensional matrix. Matrix inversion is very computationally intensive and thus becomes much slower for models having large numbers of clusters. The algorithm is said to converge when successive iterations change the proportion estimates by less than a specified tolerance.

In order to implement the Newton-Raphson algorithm, the first derivatives of the log likelihood (for the **B** matrix) and the second derivatives of the log likelihood (for the **H** matrix) are needed. These derivatives were derived for the normal mixture distribution by Hasselblad (1966). The notation and first derivatives were defined in Section 4.3.2. The first derivatives are given in Equations 4.15 – 4.17.

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 \ell}{\partial \pi^2} & \dfrac{\partial^2 \ell}{\partial \pi \partial \mu_1} & \dfrac{\partial^2 \ell}{\partial \pi \partial \mu_2} & \dfrac{\partial^2 \ell}{\partial \pi \partial \sigma_1} & \dfrac{\partial^2 \ell}{\partial \pi \partial \sigma_2} \\ \dfrac{\partial^2 \ell}{\partial \mu_1 \partial \pi} & \dfrac{\partial^2 \ell}{\partial \mu_1^2} & \dfrac{\partial^2 \ell}{\partial \mu_1 \partial \mu_2} & \dfrac{\partial^2 \ell}{\partial \mu_1 \partial \sigma_1} & \dfrac{\partial^2 \ell}{\partial \mu_1 \partial \sigma_2} \\ \dfrac{\partial^2 \ell}{\partial \mu_2 \partial \pi} & \dfrac{\partial^2 \ell}{\partial \mu_2 \partial \mu_1} & \dfrac{\partial^2 \ell}{\partial \mu_2^2} & \dfrac{\partial^2 \ell}{\partial \mu_2 \partial \sigma_1} & \dfrac{\partial^2 \ell}{\partial \mu_2 \partial \sigma_2} \\ \dfrac{\partial^2 \ell}{\partial \sigma_1 \partial \pi} & \dfrac{\partial^2 \ell}{\partial \sigma_1 \partial \mu_1} & \dfrac{\partial^2 \ell}{\partial \sigma_1 \partial \mu_2} & \dfrac{\partial^2 \ell}{\partial (\sigma_1)^2} & \dfrac{\partial^2 \ell}{\partial \sigma_1 \partial \sigma_2} \\ \dfrac{\partial^2 \ell}{\partial \sigma_2 \partial \pi} & \dfrac{\partial^2 \ell}{\partial \sigma_2 \partial \mu_1} & \dfrac{\partial^2 \ell}{\partial \sigma_2 \partial \mu_2} & \dfrac{\partial^2 \ell}{\partial \sigma_2 \partial \sigma_1} & \dfrac{\partial^2 \ell}{\partial (\sigma_2)^2} \end{bmatrix} \tag{4.28}$$

$$\mathbf{B} = \begin{bmatrix} \dfrac{\partial \ell}{\partial \pi} \\ \dfrac{\partial \ell}{\partial \mu_1} \\ \dfrac{\partial \ell}{\partial \mu_2} \\ \dfrac{\partial \ell}{\partial \sigma_1} \\ \dfrac{\partial \ell}{\partial \sigma_2} \end{bmatrix} \tag{4.29}$$

The second derivatives of the log likelihood are shown in Equations 4.31 – 4.36. They are obtained by taking the appropriate first derivatives of Equations 4.15 – 4.17. Define δ_{mk} as the Kronecker delta,

$$\delta_{mk} = \begin{cases} 1, \text{ if m} = k \\ 0, \text{ if m} \neq k \end{cases}. \tag{4.30}$$

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \mu_m \partial \mu_k} = & \sum_{i=1}^{N} \frac{-\pi_m \pi_k f_{im} f_{ik}}{\sigma_m^2 \sigma_k^2 F_i^2} \left(y_i - \mu_m \right) \left(y_i - \mu_k \right) + \\ & \delta_{mk} \sum_{i=1}^{N} \frac{\pi_k f_{ik}}{F_i} \left[\frac{\left(y_i - \mu_k \right)^2}{\sigma_k^4} - \frac{1}{\sigma_k^2} \right]\end{aligned} \tag{4.31}$$

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \sigma_m \partial \mu_k} = & \sum_{i=1}^{N} \frac{-\pi_m \pi_k f_{im} f_{ik}}{\sigma_k^2 F_i^2} \left(y_i - \mu_k \right) \left[\frac{-1}{\sigma_m} + \frac{\left(y_i - \mu_m \right)^2}{\sigma_m^3} \right] + \\ & \delta_{mk} \sum_{i=1}^{N} \frac{\pi_k f_{ik}}{F_i} \left(y_i - \mu_k \right) \left[\frac{-3}{\sigma_k^3} + \frac{\left(y_i - \mu_k \right)^2}{\sigma_k^5} \right]\end{aligned} \tag{4.32}$$

$$\frac{\partial^2 \ell}{\partial \pi_m \partial \mu_k} = \sum_{i=1}^{N} \frac{-\pi_k f_{ik}}{\sigma_k^2 F_i^2} \left(y_i - \mu_k \right) \left(f_{im} - f_{iC} \right) + \left(\delta_{mk} - \delta_{kC} \right) \sum_{i=1}^{N} \frac{f_{ik} \left(y_i - \mu_k \right)}{\sigma_k^2 F_i} \tag{4.33}$$

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \sigma_m \partial \sigma_k} = & \sum_{i=1}^{N} \frac{-\pi_m \pi_k f_{im} f_{ik}}{F_i^2} \left[\frac{\left(y_i - \mu_m \right)^2}{\sigma_m^3} - \frac{1}{\sigma_m} \right] \left[\frac{\left(y_i - \mu_k \right)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right] \\ & + \delta_{mk} \sum_{i=1}^{N} \frac{\pi_k f_{ik}}{F_i} \left(\frac{-1}{\sigma_k} \right) \left[\frac{3 \left(y_i - \mu_k \right)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right] \left[\frac{\left(y_i - \mu_k \right)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right]^2\end{aligned} \tag{4.34}$$

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \pi_m \partial \sigma_k} = & \sum_{i=1}^{N} \frac{-\pi_k f_{ik}}{F_i^2} \left(f_{im} - f_{iC} \right) \left[\frac{\left(y_i - \mu_k \right)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right] \\ & + \left(\delta_{mk} - \delta_{kC} \right) \sum_{i=1}^{N} \frac{f_{ik}}{F_i} \left[\frac{\left(y_i - \mu_k \right)^2}{\sigma_k^3} - \frac{1}{\sigma_k} \right]\end{aligned} \tag{4.35}$$

$$\frac{\partial^2 \ell}{\partial \pi_m \partial \pi_k} = \sum_{i=1}^{N} \frac{-\left(f_{ik} - f_{iC} \right) \left(f_{im} - f_{iC} \right)}{F_i^2} \tag{4.36}$$

The implementation of the NR algorithm is straightforward. However, the NR algorithm requires the observed information or Hessian matrix, which is complex to calculate for the mixture problem. The NR algorithm is also much more sensitive to starting values than the EM algorithm is. The NR algorithm sometimes returns negative estimates for σ. When this happens, we follow Aitkin and Aitkin's (1996) suggestion and change the sign of these estimates. There is also a possibility that the Hessian matrix may not be positive definite and thus is non-invertible. This possibility increases if the starting values are poor. When this happens, the EM algorithm must be used instead.

The formulas needed to implement the EM and NR algorithms for the normal mixture case are given in Equations 4.15 – 4.36. The EM/Newton-Raphson hybrid algorithm was introduced by Aitkin and Aitkin (1996) to take advantage of the best features of both the EM and NR algorithms. The algorithm is outlined in the flowchart shown in Figure 4.7.

The steps are referenced by the numbers in parentheses. The first step is to run the EM algorithm 5 times (1). This helps to ensure that the log-likelihood is non-decreasing so that the subsequent NR step will not diverge or decrease the log-likelihood. Running the EM algorithm 5 times comes from Redner and Walker's (1984) experience that 95% of the change in log-likelihood from the initial to the maximum value generally occurred in the first five EM iterations. If the EM algorithm did not converge after 5 iterations, the NR algorithm is run until it converges or the likelihood decreases (5 – 14). If the likelihood decreases, the parameter values are set to the average of their values before the likelihood decreased and their current values (11). The NR algorithm is run again with the new parameter values as starting values (13). This process is called step halving. If 5 step-halvings do not increase the log-likelihood, then the EM algorithm is run 5 more times (1). This process is repeated as shown in Figure 4.7 until convergence is obtained or a user defined maximum number of iterations is

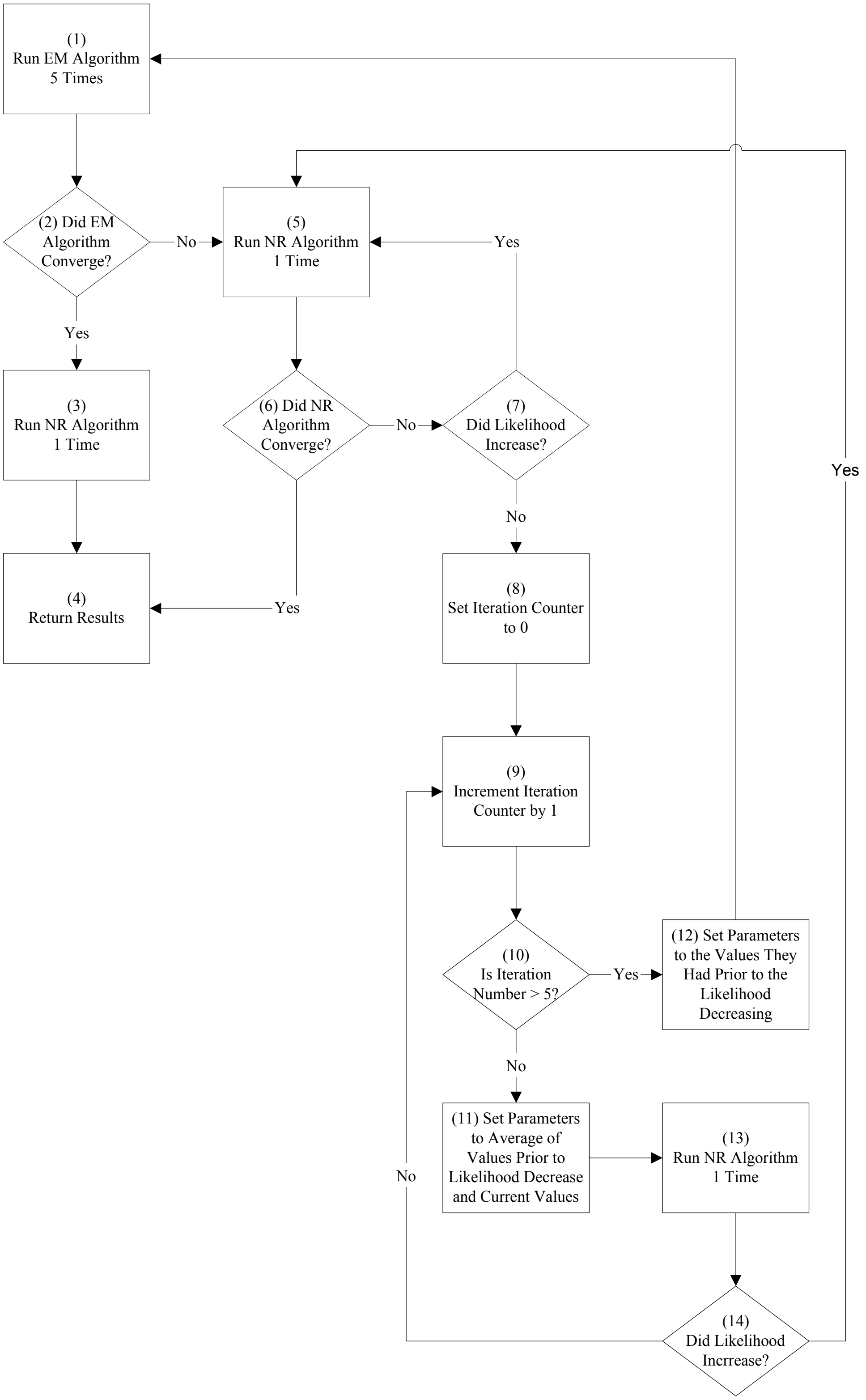


Figure 4.7: Flowchart for the EM/Newton-Raphson Hybrid Algorithm

reached. Once the EM/Newton-Raphson hybrid algorithm converges, the parameter estimates may be used to calculate confidence intervals (Section 4.3.4) and to assign observations to clusters.

One must be careful to recognize that the EM algorithm estimates the σ_k^2 parameters, while the NR algorithm estimates the σ_k parameters (as derived by Hasselblad in 1966). The σ_k^2 estimates from the EM step in the hybrid algorithm must be translated by taking the square root before using them as inputs for the NR step. Similarly, one should square the NR estimates before using them as inputs for the EM step.

The application of the EM/Newton-Raphson hybrid algorithm usually results in a significantly reduced number of EM algorithm iterations. McLachlan and Peel (2000) report that, in their experience, the hybrid algorithm converges in 50 – 70 percent of the time required for the EM algorithm to converge. However, the hybrid algorithm requires more overhead for implementation. The Hessian matrix is complex to calculate. The EM algorithm usually converges much faster than the hybrid algorithm for the univariate mixture models presented in this dissertation. The convergence time depends on many things, including the number of variables. The examples McLachlan and Peel (2000) discussed were all multivariate applications. For univariate mixtures, the EM algorithm appears to be more desirable. The hybrid algorithm is best applied to multivariate mixture problems and will not be used in this dissertation (due to the faster convergence of the EM algorithm for univariate normal mixtures). However, the 2-DCluster software package does support the hybrid algorithm.

4.3.4 Calculating Confidence Intervals

In this section, confidence intervals are derived for the posterior probability that the i^{th} observation belongs to the k^{th} cluster, or $\hat{\alpha}_{ik}$. The maximum $\hat{\alpha}_{ik}$ provides an estimate to determine which cluster an observation is assigned to. For values of the $\hat{\alpha}_{ik}$'s that are close for more than one cluster, a confidence interval may help the user to determine in which cluster to place the observation, y_i. If $m \le k$ confidence intervals for the $\hat{\alpha}_{ik}$'s overlap for a given *i*, the observation y_i could be placed in any of the *m* clusters. Thus, the confidence intervals for $\hat{\alpha}_{ik}$ help to support the use of overlapping clusters. Overlapping clusters may be desirable in some clustering situations.

The derivation of the confidence interval for $\hat{\alpha}_{ik}$ makes use of the delta method. The delta method is frequently used for finding the asymptotic variances of functions of parameters (Bickel and Doksum, 2001). Equation 4.37 is obtained by plugging the parameter estimates into Equation 4.19, or:

$$\hat{\alpha}_{ik} = \frac{\hat{\pi}_k \ \phi\left(y_i;\ \hat{\mu}_k,\ \hat{\sigma}_{\mathrm{k}}^2 \right)}{\sum_{r=1}^{C} \hat{\pi}_r \ \phi\left(y_i;\ \hat{\mu}_r,\ \hat{\sigma}_{\mathrm{r}}^2 \right)}. \tag{4.37}$$

The delta method has the form:

$$V\left(\hat{\alpha}_{ik}\right) = f'(\underline{\theta})\Sigma\left[f'(\underline{\theta})\right]^T, \tag{4.38}$$

where $\underline{\theta}$ represents the vector of unknown parameters in $\widehat{\alpha}_{ik}$, which are π_k, μ_k, and σ_k. The *T* refers to the matrix transpose operation. The first derivative of *f* with respect to these parameters is:

$$f'(\underline{\theta}) = \left[\frac{\partial \widehat{\alpha}_{ik}}{\partial \pi_k} \ \frac{\partial \widehat{\alpha}_{ik}}{\partial \mu_k} \ \frac{\partial \widehat{\alpha}_{ik}}{\partial \sigma_k} \right]. \tag{4.39}$$

The symmetric matrix Σ is:

$$\begin{bmatrix} Var(\pi_k) & Cov(\mu_k, \pi_k) & Cov(\sigma_k, \pi_k) \\ Cov(\pi_k, \mu_k) & Var(\mu_k) & Cov(\sigma_k, \mu_k) \\ Cov(\pi_k, \sigma_k) & Cov(\mu_k, \sigma_k) & Var(\sigma_k) \end{bmatrix}. \tag{4.40}$$

The derivatives needed for Equation 4.37 are derived below. The derivative of $\widehat{\alpha}_{ik}$ with respect to π_k is:

$$\begin{aligned} \frac{\partial \widehat{\alpha}_{ik}}{\partial \pi_k} &= \frac{\partial}{\partial \pi_k} \left[\frac{\pi_k \phi\left(y_i;\ \mu_\mathrm{k}, \sigma_k^2\right)}{\sum_{r=1}^{C} \pi_r \phi\left(y_i;\ \mu_\mathrm{r}, \sigma_r^2\right)} \right] \\ &= \frac{\phi\left(y_i;\ \mu_\mathrm{k}, \sigma_k^2\right) \sum_{r=1}^{C} \pi_r \phi\left(y_i;\ \mu_\mathrm{r}, \sigma_r^2\right) - \pi_k \left[\phi\left(y_i;\ \mu_\mathrm{k}, \sigma_k^2\right) \right]^2}{\left[\sum_{r=1}^{C} \pi_r \phi\left(y_i;\ \mu_\mathrm{r}, \sigma_r^2\right) \right]^2} \end{aligned} \tag{4.41}$$

$$
=\frac{\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\left[\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)-\pi_k\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\right]}{\left[\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]^2}.
$$

The derivative of $\widehat{\alpha}_{ik}$ with respect to μ_k is:

$$
\begin{aligned}
\frac{\partial\widehat{\alpha}_{ik}}{\partial\mu_k}&=\frac{\partial}{\partial\mu_k}\left[\frac{\pi_k\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)}{\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)}\right]\\
&=\left\{\pi_k\left(y_i-\mu_k\right)\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\sum_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)-\right.\\
&\qquad\left.\pi_k^2\left(y_i-\mu_k\right)\left[\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\right]^2\right\}\ /\ \left\{\sigma_k^4\left[\sum_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]^2\right\}\\
&=\frac{\pi_k\left(y_i-\mu_k\right)\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\left[\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)-\pi_k\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\right]}{\sigma_k^4\left[\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]^2}.
\end{aligned}
\tag{4.42}
$$

The derivative of $\widehat{\alpha}_{ik}$ with respect to σ_k is:

$$\begin{aligned}
\frac{\partial\widehat{\alpha}_{ik}}{\partial\sigma_k} &= \frac{\partial}{\partial\sigma_k}\left[\frac{\pi_k\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)}{\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)}\right] \\
&= \left(\frac{\left(y_i-\mu_k\right)^2}{\sigma_k^2}-1\right)\left(\left\{\pi_k\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)\sum_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right.\right. \\
&\quad \left.\left.-\pi_k^2\left[\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]^2\right\} \ / \ \left\{\sigma_k\left[\sum_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]^2\right\}\right) \\
&= \left(\frac{\pi_k\phi\left(y_i;\ \mu_{\mathrm{k}},\sigma_k^2\right)}{\sigma_k}\right)\left(\frac{\left(y_i-\mu_k\right)^2}{\sigma_k^2}-1\right) \\
&\quad \left(\frac{\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)-\pi_k\left[\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]}{\left[\sum\limits_{r=1}^{C}\pi_r\phi\left(y_i;\ \mu_{\mathrm{r}},\sigma_r^2\right)\right]^2}\right).
\end{aligned} \tag{4.43}$$

Note that all of the parameter estimates are required to solve Equations 4.41 – 4.43. The elements of Σ are obtained from the corresponding elements of $I^{-1}\left(\underline{\theta}\right)$, where *I* is the Fisher information matrix. This is calculated by taking the negative of the inverse of the Hessian matrix defined in Equation 4.28. The Hessian matrix given in Equation 4.28 does not include the estimate for π_C since it can be found from the constraint $\pi_C = 1-\sum\limits_{k=1}^{C-1}\pi_k$. These estimates are asymptotically normal, due to the properties of maximum likelihood estimation. The missing elements of the Hessian matrix related to π_C are obtained by plugging the estimated values into the appropriate derivatives. Once the algorithm terminates, estimates for all of the parameter values are available.

A confidence interval for posterior probability, α_{ik}, may be calculated as:

$$\widehat{\alpha}_{ik} \pm z_{1-\alpha/2}\sqrt{Var(\widehat{\alpha}_{ik})}, \qquad (4.44)$$

where $z_{1-\alpha/2}$ is the usual $100(1-\alpha/2)$ percentile of the standard normal distribution. For example, to obtain 95 percent confidence limits, $\alpha = 0.05$ and $z_{1-\alpha/2} = 1.96$

Confidence intervals may also be calculated for the cluster specific parameters. For the calculation of the confidence intervals, the covariances between the parameters are set to zero. This is justified since the observed covariances are on the order of 10^{-16} for these data. This is not necessarily true for all microarray data sets. Equation 4.45 gives the confidence interval formula for the group proportion, π_k, as:

$$\widehat{\pi}_k \pm z_{1-\alpha/2}\sqrt{Var(\widehat{\pi}_k)}. \qquad (4.45)$$

Equation 4.46 gives the confidence interval formula for the group mean, μ_k, as:

$$\widehat{\mu}_k \pm z_{1-\alpha/2}\sqrt{Var(\widehat{\mu}_k)}. \qquad (4.46)$$

Equation 4.47 gives the confidence interval formula for the group standard deviation, σ_k.

$$\widehat{\sigma}_k \pm z_{1-\alpha/2}\sqrt{Var(\widehat{\sigma}_k)}. \qquad (4.47)$$

By the invariance property of the MLE, the confidence intervals for the group variance, σ_k^2, are found by squaring the estimate obtained for σ_k and applying the usual confidence interval formula.

4.3.5 Comparing Models With Different Numbers of Clusters

In practice, the number of clusters is usually unknown. In this case, several models can be fitted using different numbers of clusters. To help in evaluating which models fit the data best, three statistics are useful. These statistics are the log likelihood (ℓ), the Akaike information criterion (AIC), and the Bayesian information criterion (BIC).

The usual way of evaluating models is to select the model having the greatest log likelihood. However, the likelihood tends to increase with the number of clusters (McLachlan and Peel, 2000). The greatest log likelihood value occurs when every observation is correctly contained in its own cluster. The log likelihood does not adjust for the number of parameters in the model. Tests such as the likelihood ratio test (LRT) (or scree plots) could be applied to select an appropriate number of clusters if the models were nested (Bickel and Doksum, 2001). However, mixture models for clusters are not nested because the clusters may not contain the same members when they are subdivided.

Akaike (1974) proposed another method for evaluating model fit known as the Akaike information criterion (AIC). The optimal model according to the AIC is the one that has the AIC value closest to zero. The AIC adjusts the likelihood for the number of parameters, but still tends to favor models with large numbers of clusters (McLachlan and Peel, 2000). The AIC value is calculated as:

$$AIC = -2\ell + 2d, \tag{4.48}$$

where ℓ is the log likelihood value and d is the degrees of freedom for the model. Despite its bias in selecting models with larger numbers of clusters, the AIC is widely used for evaluating the number of clusters present in mixture models (Bozdogan and Sclove, 1984; Sclove, 1987).

Schwarz (1978) proposed a method for evaluating model fit known as the Bayesian information criterion (BIC). The model having the BIC value closest to zero is chosen as the best fitting model. The BIC also adjusts for the number of model parameters. The adjustment for the number of parameters is multiplied by the log of the sample size. This improves on the AIC which adjusts by a constant which does not depend on the sample size. The BIC does not seem to favor models having large number of clusters (McLachlan and Peel, 2000). The BIC is calculated as:

$$BIC = -2\ell + d\log n, \tag{4.49}$$

where ℓ is the log likelihood value, d is the degrees of freedom for the model, and n is the sample size. The general form of the BIC uses a prior but the expression in Equation 4.49 assumes that the prior has little effect and can be ignored. McLachlan and Peel (2000) performed simulations and found that the BIC performs better than the AIC for choosing mixture models having the correct number of clusters.

4.3.6 Analysis of Pearson Crab Data

Karl Pearson (1894) fitted a two component univariate normal mixture model to a set of measurements on the ratio of forehead to body length on 1,000 crabs. The crab data was first reported by Weldon (1892). The data as reported in the article are shown in Table 4.2.

Table 4.2: Pearson Crab Data

N	Interval Range	N	Interval Range
1	0.580 – 0.583	74	0.640 – 0.643
3	0.584 – 0.587	84	0.644 – 0.647
5	0.588 – 0.591	86	0.648 – 0.651
2	0.592 – 0.595	96	0.652 – 0.655
7	0.596 – 0.599	85	0.656 – 0.659
10	0.600 – 0.603	75	0.660 – 0.663
13	0.604 – 0.607	47	0.664 – 0.667
19	0.608 – 0.611	43	0.668 – 0.671
20	0.612 – 0.615	24	0.672 – 0.675
25	0.616 – 0.619	19	0.676 – 0.679
40	0.620 – 0.623	9	0.680 – 0.683
31	0.624 – 0.627	5	0.684 – 0.687
60	0.628 – 0.631	0	0.688 – 0.691
62	0.632 – 0.635	1	0.692 – 0.695
54	0.636 – 0.639		

The 1,000 observations are divided into 29 categories of width 0.004. The individual measurements were not provided. In order to more accurately represent the original data, each observation *X* was generated in the following manner:

$$X_i = \mu_j \pm U * \sqrt{0.004}, \qquad (4.50)$$

where $i = 1,\ldots,1000$ indicates the observation number, μ_j indicates the mean of the j^{th} class interval, $j = 1,\ldots,29$, and *U* indicates a uniform random number on $(0,1)$. The class variance was set at 0.004. A two component normal mixture model was fitted to this data. The results are shown in Table 4.3. Pearson analyzed these data and illustrated two component normal mixture models using moment estimators.

Table 4.3: Results of Pearson Crab Data Analysis

Parameter	Starting Value	Estimate	95% Confidence Interval
π_1	0.500	0.567	(0.404, 0.731)
π_2	0.500	0.433	(0.269, 0.595)
μ_1	0.500	0.617	(0.606, 0.629)
μ_2	0.600	0.680	(0.671, 0.689)
σ_1^2	0.010	0.001	(0.000*, 0.004)
σ_2^2	0.020	0.001	(0.000*, 0.003)

*Calculated value was negative.

The model converged in 572 EM algorithm iterations. The log likelihood was 1779.85, the AIC was -3549.71, and the BIC was -3525.16. Pearson reported proportions of 0.585 and 0.415 for the two groups using the method of moments. The method of moments is not feasible for fitting models with large numbers of groups due to the complexity of the underlying formulas. The method of moments yields unbiased estimators and is consistent in the sense that the estimates should approach the true values as the sample size increases (Bickel and Doksum, 2001). However, maximum likelihood estimation (MLE) has more desirable properties. MLE's are asymptotically unbiased, have the minimum variance, and are asymptotically normal. MLE's can be biased for small samples. In the crab example, the sample size is 1,000 and the method of moments results reported by Pearson involved a large number of hand calculations. MLE's are preferable and are now computationally tractable.

Pearson interpreted the presence of two components as evidence that the sample contained two species of crabs. The two distributions show up as modes in the density shown in Figure 4.8.

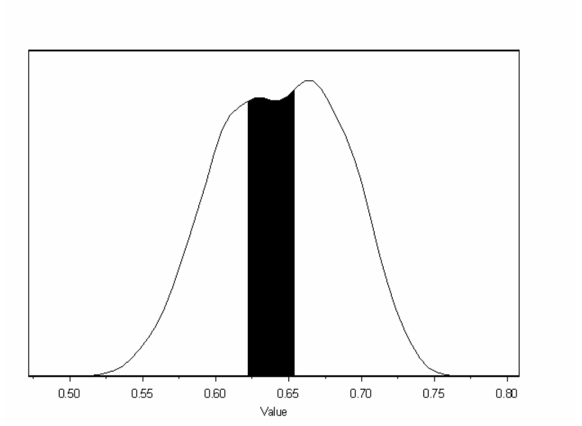


Figure 4.8: Pearson Crab Data Density Plot

Notice that there is some overlap between the two distributions. Estimation is more difficult for observations falling into the shaded (overlapping) area. This difficulty is reflected by the posterior probabilities both being close to 0.5 for some observations. For example, for one observation the posterior probability of belonging to cluster 1 is 0.52, while the posterior probability of belonging to cluster 2 is 0.48. Determining which cluster to assign the observation to is difficult. One may wish to allow overlapping clusters by including such an observation in both clusters.

4.3.7 Analysis of Simulated Data

Simulated 15 Component Normal Mixtures

Clustering applications frequently involve more than two clusters. For illustration purposes, three 15 component normal mixture distributions are simulated and analyzed. The first simulation has different proportions, well separated means, and variances that are in a fixed ratio to the proportions. The second simulation has equal proportions and different means and variances. The third simulation has equal proportions and variances but different means. SAS code for simulating *C* component normal mixture distributions is given in Appendix 4.2.

The data for the first mixture distribution are simulated using the parameter values shown in Table 4.4.

Table 4.4: Parameters for 15 Component Normal Mixture Model with Different Proportions, Different Means, and Variances in a Fixed Ratio to the Proportions

Cluster Number	Proportion	Mean	Variance
1	0.200	150.000	15.000
2	0.100	140.000	7.500
3	0.100	130.000	7.500
4	0.050	120.000	3.750
5	0.050	110.000	3.750
6	0.050	100.000	3.750
7	0.050	90.000	3.750
8	0.050	80.000	3.750
9	0.050	70.000	3.750
10	0.050	60.000	3.750
11	0.050	50.000	3.750
12	0.050	40.000	3.750
13	0.050	30.000	3.750
14	0.050	20.000	3.750
15	0.050	10.000	3.750

For this simulation, the cluster means are generated according to the formula $\mu_k = \mu_{k-1} - 10$, where $\mu_1 = 150$. The cluster variances are generated according to the formula $\sigma_k^2 = 75\pi_k$. This makes the largest variance equal to 15 (when the proportion is 0.2) and the smallest variance equal to 3.75 (when the proportions are 0.05). The variances must change in order for the proportions to differ. Even for such well separated clusters, the normal mixture density is complicated as shown in Figure 4.9.

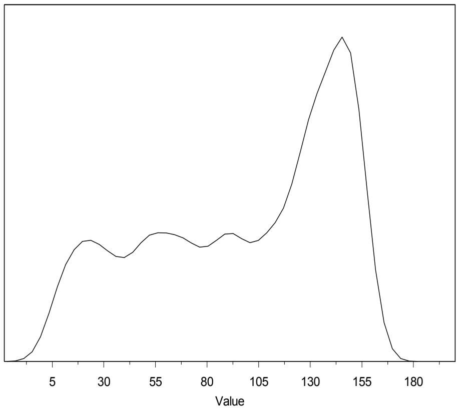


Figure 4.9: 15 Component Normal Mixture Density Plot for First Simulation

Notice that the large mode on the right is due to a large percentage of the observations being concentrated in this region. There were 1,000 observations simulated. A 15 component one dimensional mixture distribution was fitted. The starting values were generated from a k-means cluster analysis. These starting values are shown in Table 4.5.

Table 4.5: Starting Values for First Simulation of 15 Component Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.100	15.425	32.802
2	0.072	33.597	21.432
3	0.066	49.489	9.681
4	0.045	60.289	2.433
5	0.123	141.248	7.844
6	0.116	130.476	5.288
7	0.051	69.376	4.235
8	0.054	120.797	3.332
9	0.017	77.423	2.669
10	0.060	110.506	4.217
11	0.026	81.136	1.430
12	0.038	89.285	1.080
13	0.023	92.125	1.062
14	0.170	151.267	9.368
15	0.039	99.970	3.507

The model converged in 38 seconds and required 1,057 EM algorithm iterations. The log likelihood for the model was -4782.91, the AIC was 9653.83, and the BIC was 9869.77. The parameter estimates are reported in Table 4.6.

Table 4.6: Fit Results for First Simulation of 15 Component Normal Mixture Model

Actual Parameters			Estimated Parameters		
Proportion	Mean	Variance	Proportion	Mean	Variance
0.200	150.000	15.000	0.200	150.204	14.668
0.100	140.000	7.500	0.096	140.028	6.987
0.100	130.000	7.500	0.115	130.215	5.810
0.050	120.000	3.750	0.053	120.531	2.325
0.050	110.000	3.750	0.060	110.489	4.261
0.050	100.000	3.750	0.039	99.877	3.801
0.050	90.000	3.750	0.003	93.441	0.007
0.050	80.000	3.750	0.057	90.126	2.328
0.050	70.000	3.750	0.031	80.628	2.417
0.050	60.000	3.750	0.001	77.833	1.193
0.050	50.000	3.750	0.053	69.594	4.870
0.050	40.000	3.750	0.045	60.357	2.453
0.050	30.000	3.750	0.050	50.835	1.691
0.050	20.000	3.750	0.104	33.309	73.188
0.050	10.000	3.750	0.084	14.827	36.864

Notice in Table 4.6 that the large clusters were estimated reasonably accurately. The best case cluster labeling algorithm described in chapter 2 is applied. The kappa statistic is 0.792, the weighted kappa statistic is 0.939, the Rand index is 0.977, and the adjusted Rand index is 0.864. All of these statistics (discussed in Chapter 2) indicate good agreement between the mixture model clustering and the true simulated clusters. Some of the small clusters were incorrectly estimated. Increasing the sample size should improve estimation since the algorithms used are based on asymptotic theory. In practice, we suggest removing the observations contained in large clusters and re-clustering the remaining observations. The idea behind this recommendation is that large clusters are likely to contain a high amount of noise which makes it hard to extract possible signals of interest. The mean estimation is generally more accurate than the variance estimation. This is due to the difficulty in discerning where the tails of the normal component distributions lie in the mixture distribution.

The parameters for the second 15 component normal mixture distribution simulation are given in Table 4.7. The proportions are equal, the means are generated the same way as in the first simulation, and the variances are generated according to the formula $\sigma_k^2 = \sigma_{k-1}^2 - 1$, where $\sigma_1^2 = 15$.

Table 4.7: Parameters for 15 Component Normal Mixture Model with Equal Proportions, Different Means, and Different Variances

Cluster Number	Proportion	Mean	Variance
1	0.067	150.000	15.000
2	0.067	140.000	14.000
3	0.067	130.000	13.000
4	0.067	120.000	12.000
5	0.067	110.000	11.000
6	0.067	100.000	10.000
7	0.067	90.000	9.000
8	0.067	80.000	8.000
9	0.067	70.000	7.000
10	0.067	60.000	6.000
11	0.067	50.000	5.000
12	0.067	40.000	4.000
13	0.067	30.000	3.000
14	0.067	20.000	2.000
15	0.067	10.000	1.000

The normal mixture density for the second simulation is shown in Figure 4.10. The density is complex and it is difficult to see more than a handful of modes (there should be 15 total modes).

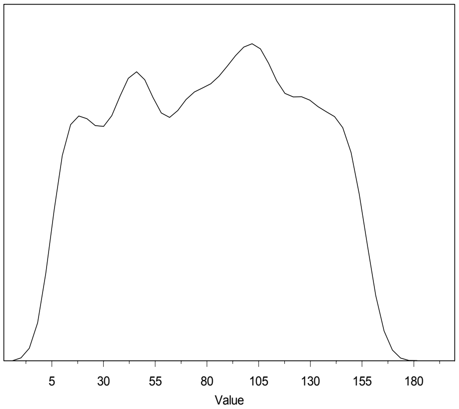


Figure 4.10: 15 Component Normal Mixture Density Plot for Second Simulation

There were 1,000 observations simulated. A 15 component one dimensional mixture distribution was fitted. The starting values were generated from a k-means cluster analysis. These starting values are shown in Table 4.8.

Table 4.8: Starting Values for Second Simulation of 15 Component Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.067	10.075	2.334
2	0.109	24.414	25.000
3	0.078	40.163	7.198
4	0.072	50.158	4.077
5	0.069	140.790	8.761
6	0.075	128.949	8.916
7	0.052	60.219	4.771
8	0.055	119.320	7.228
9	0.062	69.886	3.620
10	0.070	109.645	4.941
11	0.046	82.174	3.284
12	0.031	76.867	2.968
13	0.075	100.713	5.189
14	0.065	151.983	12.043
15	0.074	90.986	6.001

The model converged in 42 seconds and required 1,455 EM algorithm iterations. The log likelihood for the model was -4904.65, the AIC was 9897.30, and the BIC was 10113.24. The parameter estimates are reported in Table 4.9.

Table 4.9: Fit Results for Second Simulation of 15 Component Normal Mixture Model

Actual Parameters			Estimated Parameters		
Proportion	Mean	Variance	Proportion	Mean	Variance
0.067	150.000	15.000	0.066	151.624	15.249
0.067	140.000	14.000	0.069	140.703	12.819
0.067	130.000	13.000	0.029	130.621	3.883
0.067	120.000	12.000	0.105	122.561	34.741
0.067	110.000	11.000	0.068	109.403	7.397
0.067	100.000	10.000	0.070	100.658	5.464
0.067	90.000	9.000	0.082	90.720	9.381
0.067	80.000	8.000	0.010	82.409	0.053
0.067	70.000	7.000	0.057	79.386	7.676
0.067	60.000	6.000	0.066	70.087	5.241
0.067	50.000	5.000	0.050	60.195	4.508
0.067	40.000	4.000	0.074	50.076	4.922
0.067	30.000	3.000	0.065	40.808	2.715
0.067	20.000	2.000	0.125	25.090	38.936
0.067	10.000	1.000	0.063	9.863	0.968

Notice in Table 4.9 that many cluster parameters were estimated accurately. The best case cluster labeling algorithm described in chapter 2 is applied. The kappa statistic is 0.484, the weighted kappa statistic is 0.829, the Rand index is 0.968, and the adjusted Rand index is 0.757. All of these statistics (discussed in Chapter 2) indicate good agreement between the mixture model clustering and the true simulated clusters. The agreement is slightly worse than that for the first simulated 15 component normal mixture model. Some of the observations were inappropriately combined. When the mixture distribution is such that the number of observations in each cluster is large, estimation should improve since the algorithms used are based on asymptotic theory. Thus, increasing the sample size may improve the estimation.

Mixture models may not be appropriate for applications with small numbers of observations, as the estimation could deteriorate. The mean estimation is generally more accurate than the variance estimation. This is due to the difficulty in discerning where the tails of the normal component distributions lie in the mixture distribution.

The parameters for the third 15 component normal mixture distribution simulation are given in Table 4.10. The proportions are equal, the means are generated the same way as in the first and second simulations, and the variances are equal.

Table 4.10: Parameters for 15 Component Normal Mixture Model with Equal Proportions, Different Means, and Equal Variances

Cluster Number	Proportion	Mean	Variance
1	0.067	150.000	5.000
2	0.067	140.000	5.000
3	0.067	130.000	5.000
4	0.067	120.000	5.000
5	0.067	110.000	5.000
6	0.067	100.000	5.000
7	0.067	90.000	5.000
8	0.067	80.000	5.000
9	0.067	70.000	5.000
10	0.067	60.000	5.000
11	0.067	50.000	5.000
12	0.067	40.000	5.000
13	0.067	30.000	5.000
14	0.067	20.000	5.000
15	0.067	10.000	5.000

The normal mixture density for the second simulation is shown in Figure 4.11. The density is complex and it is difficult to see more than a handful of modes (there should be 15 total modes).

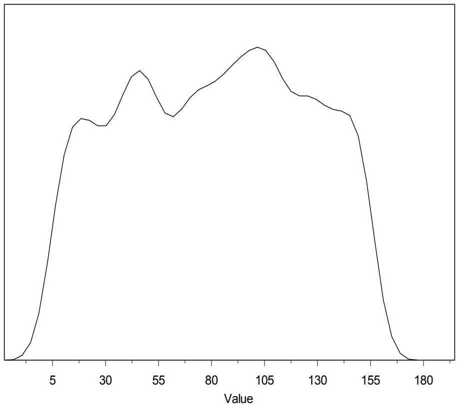


Figure 4.11: 15 Component Normal Mixture Density Plot for Third Simulation

Figure 4.11 looks virtually identical to Figure 4.10 because only the simulation variances have changed for the third simulation. There were 1,000 observations simulated. A 15 component one dimensional mixture distribution was fitted. The starting values were generated from a k-means cluster analysis. These starting values are shown in Table 4.11.

Table 4.11: Starting Values for Third Simulation of 15 Component Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.071	10.243	7.813
2	0.093	23.791	19.959
3	0.086	38.90	13.097
4	0.072	49.662	4.382
5	0.091	132.30	18.443
6	0.061	120.62	5.039
7	0.020	57.072	2.873
8	0.036	61.349	1.873
9	0.063	69.953	2.697
10	0.079	109.842	4.593
11	0.074	100.250	3.283
12	0.072	90.542	3.234
13	0.026	76.909	3.033
14	0.407	147.783	22.101
15	0.049	81.587	2.556

The model converged in 20 seconds and required 68 EM algorithm iterations. The log likelihood for the model was -4828.25, the AIC was 9716.50, and the BIC was 9863.73. A common variance for all of the clusters was estimated. This should be done only if the researcher has no reason to suspect that the variances for the groups are different. The parameter estimates are reported in Table 4.12.

Table 4.12: Fit Results for Third Simulation of 15 Component Normal Mixture Model

Actual Parameters			Estimated Parameters		
Proportion	Mean	Variance	Proportion	Mean	Variance
0.067	150.000	5.000	0.073	150.630	4.510
0.067	140.000	5.000	0.061	140.126	4.510
0.067	130.000	5.000	0.067	129.589	4.510
0.067	120.000	5.000	0.059	120.361	4.510
0.067	110.000	5.000	0.078	109.825	4.510
0.067	100.000	5.000	0.074	100.308	4.510
0.067	90.000	5.000	0.073	90.450	4.510
0.067	80.000	5.000	0.068	80.316	4.510
0.067	70.000	5.000	0.069	70.248	4.510
0.067	60.000	5.000	0.052	60.205	4.510
0.067	50.000	5.000	0.072	50.115	4.510
0.067	40.000	5.000	0.071	40.754	4.510
0.067	30.000	5.000	0.051	30.516	4.510
0.067	20.000	5.000	0.064	20.380	4.510
0.067	10.000	5.000	0.066	9.821	4.510

The results in Table 4.12 are very accurate. This is due in part to estimating a common variance. Only $2C$ parameters need to be estimated, as opposed to the usual $3C-1$. For this case, if the variances are estimated separately, the parameter estimates are not as accurate, nor is the BIC as small. The best case cluster labeling algorithm described in chapter 2 is applied. The kappa statistic is 0.971, the weighted kappa statistic is 0.994, the Rand index is 0.993, and the adjusted Rand index is 0.943. All of these statistics (discussed in Chapter 2) indicate very good agreement between the mixture model clustering and the true simulated clusters.

The usual approach to simulation is to generate a number of data sets, perform estimation separately for each one, and to report the aggregate parameter estimates along with their standard errors. Such an approach takes the randomness of simulated data into account and may give a more accurate representation of how well the algorithm is reproducing the simulated parameters. However, as discussed in Chapter 3, there is no inherent labeling for the clusters. Thus, there is no way to accurately match the clusters from the different simulations in order to report aggregate parameter estimates. This limitation is common to all clustering techniques.

However, the means (Equation 4.6) and variances (Equation 4.10) of *Y* for the mixture distribution are comparable for multiple runs of simulated data. We simulated 100 sets of 1,000 observations for each of the three cases discussed above. The same seed values were used throughout.

For data simulated using the parameter values reported in Table 4.4, Figure 4.12 shows a scatter plot of the observed means of *Y*.

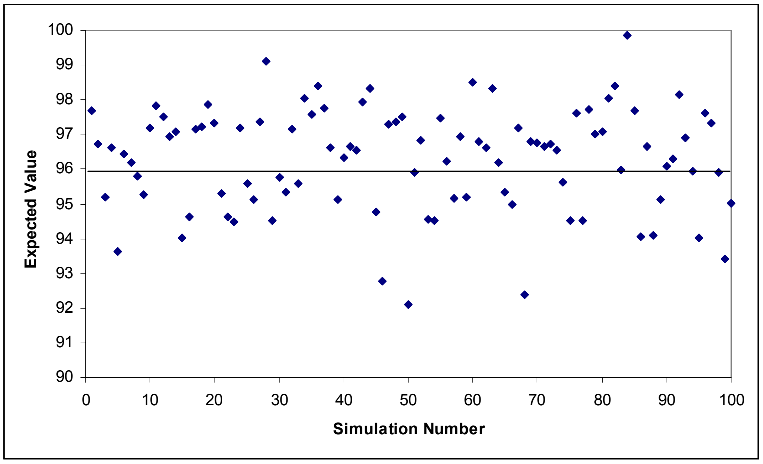


Figure 4.12: Scatter Plot of the Means of *Y* for Simulation 1 with 100 Runs

The actual mean was 96 and is indicated by the horizontal line in Figure 4.12. The mean of the 100 simulations was 96.30 with a standard error of 1.48. Figure 4.13 shows a scatter plot of the variances for *Y*.

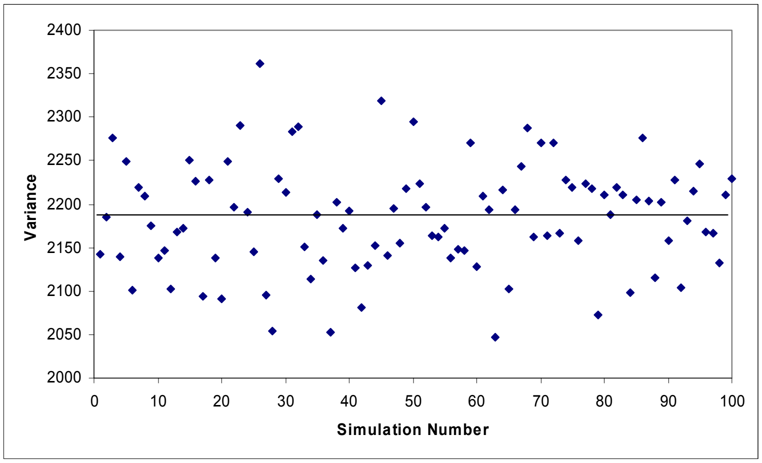


Figure 4.13: Scatter Plot of Variances of *Y* for Simulation 1 with 100 Runs

The actual variance was 2190.75 and is indicated by the horizontal line in Figure 4.13. The variance for the 100 simulations was 2184.28 with a standard error of 61.75. The simulated data appears to be representative of the artificially constructed population.

For data simulated using the parameter values reported in Table 4.7, Figure 4.14 shows a scatter plot of the observed means of *Y*. The actual mean was 80 and is indicated by the horizontal line in Figure 4.14. The mean of the 100 simulations was 80.28 with a standard error of 1.36. Figure 4.15 shows a scatter plot of the variances for *Y*.

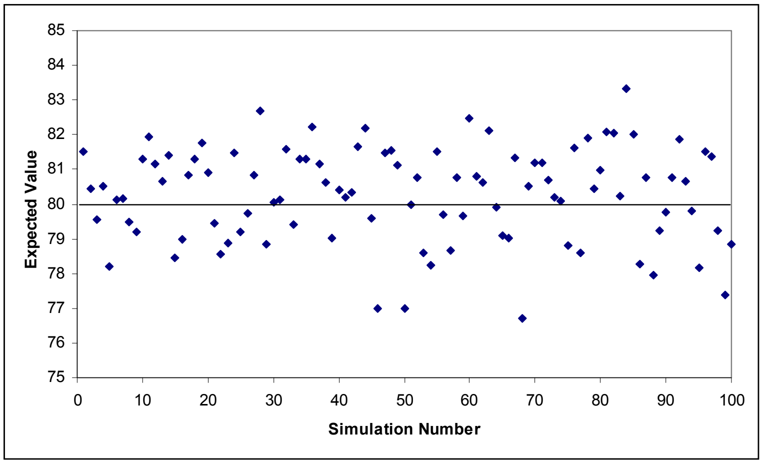


Figure 4.14: Scatter Plot of the Means of *Y* for Simulation 2 with 100 Runs

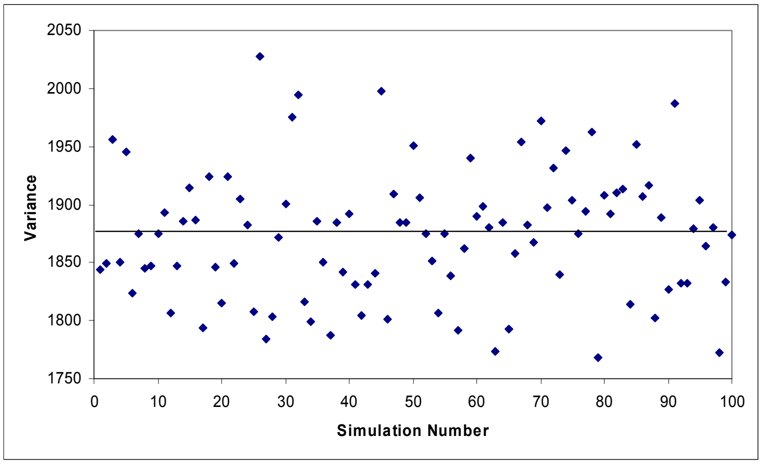


Figure 4.15: Scatter Plot of Variances of *Y* for Simulation 2 with 100 Runs

The actual variance was 1874.67 and is indicated by the horizontal line in Figure 4.15. The variance for the 100 simulations was 1873.43 with a standard error of 55.30. The simulated data appears to be representative of the artificially constructed population.

For data simulated using the parameter values reported in Table 4.10, Figure 4.16 shows a scatter plot of the observed means of *Y*.

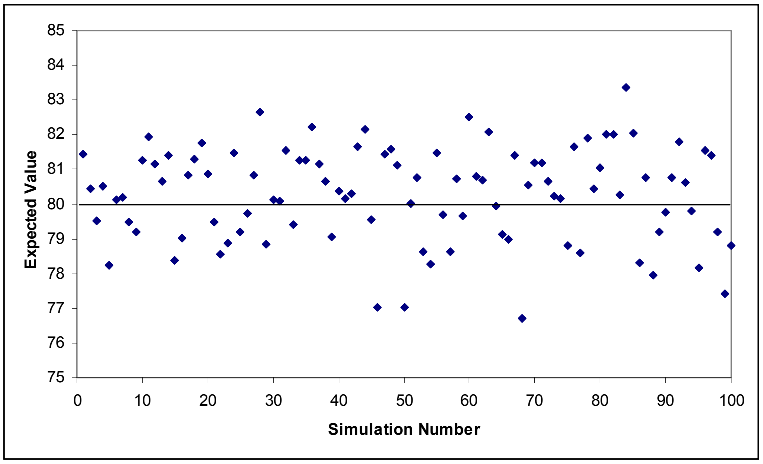


Figure 4.16: Scatter Plot of the Means of *Y* for Simulation 3 with 100 Runs

The actual mean was 80 and is indicated by the horizontal line in Figure 4.16. The mean of the 100 simulations was 80.28 with a standard error of 1.35. Figure 4.16 is very similar to Figure 4.14 because the only simulation parameters that changed were the variances. Figure 4.17 shows a scatter plot of the variances for *Y*.

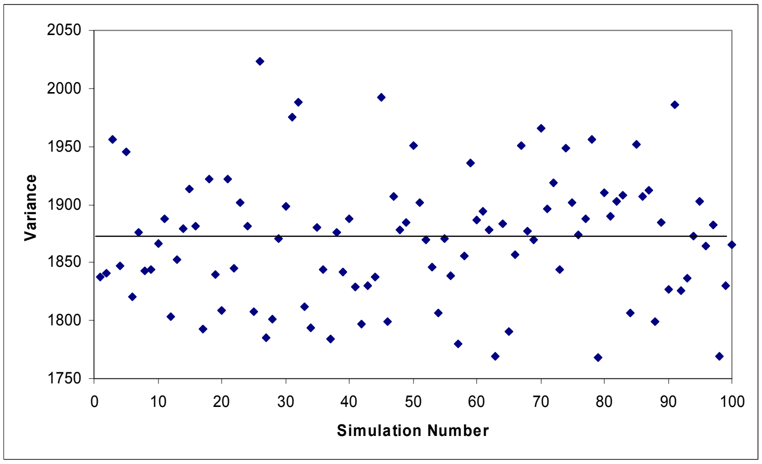


Figure 4.17: Scatter Plot of Variances of *Y* for Simulation 3 with 100 Runs

The actual variance was 1871.67 and is indicated by the horizontal line in Figure 4.17. The variance for the 100 simulations was 1870.36 with a standard error of 55.30. The simulated data appears to be representative of the artificially constructed population.

The model fit criteria described in Section 4.3.5 were applied to the first simulated 15 component normal mixture model described above. Table 4.4 gives the simulation parameter values. Assuming that the number of clusters is unknown, models ranging from 10 clusters to 20 clusters were fitted. The log likelihood, AIC, and BIC values are given in Table 4.13.

Table 4.13: Fit Statistics for Normal Mixture Models with Different Cluster Sizes

Number of Clusters	Log Likelihood	AIC	BIC
10	-4863.53	9785.06	9927.38
11	-4844.06	9752.11	9909.16
12	-4831.63	9733.25	9905.02
13	-4798.77	9673.53	9872.03
14	-4808.58	9699.15	9900.37
15	-4782.91	9653.83	**9869.77**
16	-4781.74	9657.49	9888.15
17	-4783.03	9666.06	9911.44
18	-4804.13	9714.27	9974.38
19	**-4761.42**	**9634.84**	9909.68
20	-4772.66	9663.33	9952.89

The bolded values in Table 4.13 indicate the models chosen according to each criterion. Notice that the BIC is the only criterion that selected the model with the correct number of clusters. As discussed in Section 4.3.5, the BIC is the recommended statistic for evaluating model fit. Figure 4.18 is a graphical representation of Table 4.13.

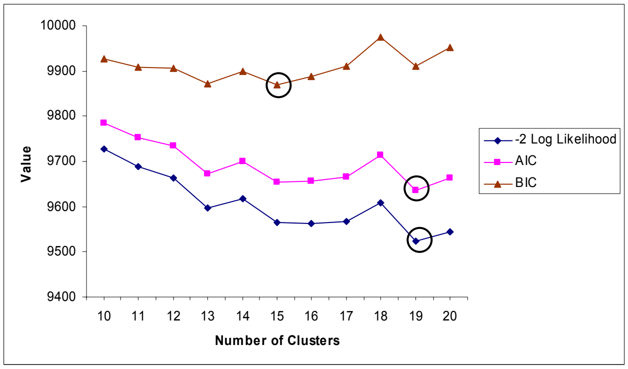


Figure 4.18: Plot of the -2 Log Likelihood, AIC, and BIC Values for 15 Component Normal Mixture

Visual inspection of Figure 4.18 confirms that the minimal BIC value occurs for the 15 cluster model, while the minimum -2 log likelihood and AIC values occur for the 19 cluster model (these values are circled in Figure 4.18). According to the BIC criterion, the 13 cluster model is also a potential candidate when choosing a final model since its BIC value is close to the value for the 15 cluster model.

4.3.8 Analysis of Ross *et al.* (2000) Data Set

Clustering Cell Lines

The Ross *et al.* (2000) data is described in Section 4.2. Complete information is available on 6,165 genes and 60 cell lines. Since the cell lines should group into 9 tumor types (Table 4.1), clustering is initially performed across the cell lines in order to compare the clustering results to the actual known tumor types. The clusters are compared using methods discussed in Chapter 3.

As discussed in Chapter 1, two dimensional data must be collapsed in order to cluster across one dimension. For this analysis, the data were collapsed by taking the mean of the 6,165 gene expression values for each of the 60 cell lines. This process ignores the variability of the observations across genes and fails to use the knowledge that all of the observations for a single cell line come from the same distribution. This problem can be setup as a repeated measures normal mixture problem. Fitting such a model requires extending the likelihood (Equation 4.12). The repeated measures problem is more completely defined in Section 6.2 and is left for future study. The density plot is given in Figure 4.19. One can readily see that there are at least three or four modes present in the data.

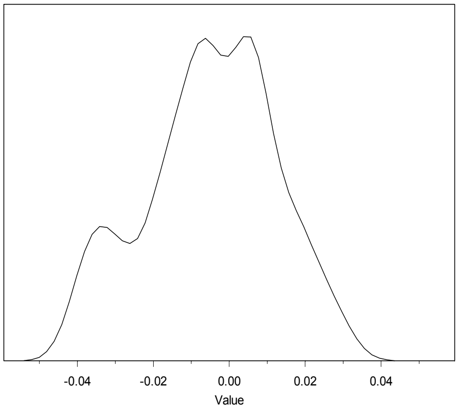


Figure 4.19: Density Plot for Cell Lines in Ross Data

In order to evaluate the normal mixture model for clustering, no assumptions were made regarding the number of clusters that are present. Normal mixture models with different numbers of clusters were fitted. The starting values were generated using a k-means cluster analysis. The fit statistics for these models are reported in Table 4.14.

Table 4.14: Fit Statistics for Clustering Across Cell Lines in Ross Data Set

Number of Clusters	Log Likelihood	AIC	BIC
2	163.70	-317.40	-306.93
3	164.09	-312.18	-295.43
4	170.09	-318.17	-295.14
5	171.30	-314.60	-285.28
6	173.42	-312.84	-277.23
7	172.45	**-304.89**	-263.00
8	175.77	-305.54	-257.37
9	179.59	-307.19	**-252.73**
10	**186.55**	-315.11	-254.37

Models having more than 10 clusters did not converge. This is likely due to the small sample size of 60. The estimation formulas are asymptotic and may not be working properly for this small sample.

Figure 4.20 is a graphical representation of Table 4.14. The bolded values in Table 4.14 and the circled values in Figure 4.20 indicate the optimal values for each of the three model fit statistics. For negative AIC and BIC values, the values closest to zero are chosen as representing the best fitting models. For the -2 log likelihood value, the smallest value is chosen. Once again, only the BIC fit statistic resulted in choosing the "right" 9 cluster model.

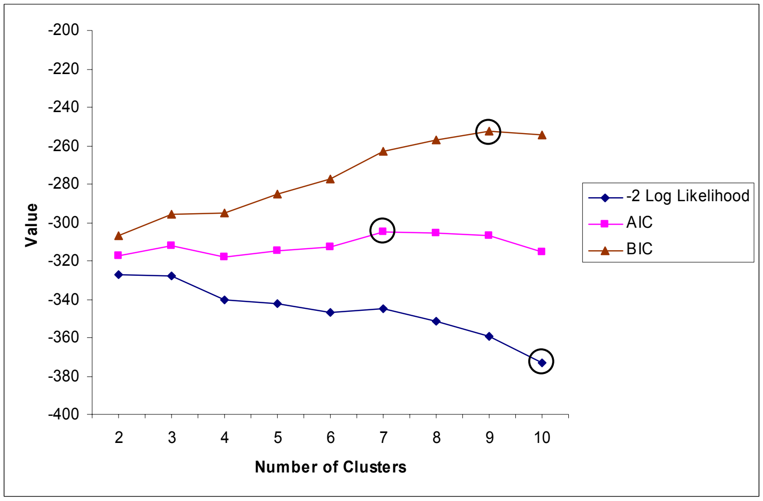


Figure 4.20: Plot of the -2 Log Likelihood, AIC, and BIC Values for Clustering Across Cell Lines Using the Ross Data

The 9 cluster model is chosen as the best fitting model. This model converged in 127 EM algorithm iterations in 36 seconds. The log likelihood value for this model was 179.59, the AIC was -307.19, and the BIC was -252.73. The parameter estimates are reported in Table 4.15.

Table 4.15: Parameter Estimates for Clustering Ross Data Across Cell Lines

Number in Cluster	Parameter Estimates		
	Proportion	Mean	Variance
12	0.195	0.006	0.0000013
6	0.101	-0.036	0.0000049
4	0.067	0.020	0.0000014
3	0.049	-0.029	0.0000007
2	0.033	0.028	0.0000007
14	0.232	-0.006	0.0000046
11	0.185	-0.016	0.0000142
4	0.071	0.002	0.0000019
4	0.067	0.014	0.0000016

Since the actual tumor types are known for the 60 cell lines, the cluster results can be verified using the methods described in Chapter 3. The "best case" cluster labeling algorithm is used using the results from the 9 cluster model. The kappa statistic is 0.188, the weighted kappa statistic is 0.111, the Rand index is 0.780, and the adjusted Rand index is 0.007. These statistics indicate that the agreement between the normal mixture model clustering and the actual cell line clusters is quite poor. This may be due to the asymptotics not working properly with a sample size of 60. Collapsing the data by taking the mean of 6,165 genes for each cell line could be problematic. It would be better to approach this analysis as a repeated measures problem. The repeated measures problem is not covered in this dissertation and is recommended as a future research problem in Section 6.2.

Clustering Genes Using Filtered Data

Proper filtering of microarray data is important in order to reduce the amount of noise present in the data. Genes which have a mean expression level close to zero and expression levels with variances close to zero are not informative and should be removed from the analysis. This method requires the selection of a threshold for the variances and the absolute values of the means. The absolute values of the means are used because the goal is to find genes with means closest to zero, regardless of whether they are positive or negative. The means and variances are calculated across the 60 cell lines for each gene. If the variances and the absolute values of the means for a gene are both below these thresholds, the gene is removed. The thresholds may be adjusted in order to change the stringency of the filter to suit the application. This filtering method was applied in this chapter, and other methods of filtering are discussed in Chapter 2.

The initial step in selecting thresholds is to plot the means and variances for all of the genes. These plots are given in Figures 4.21 and 4.22, respectively.

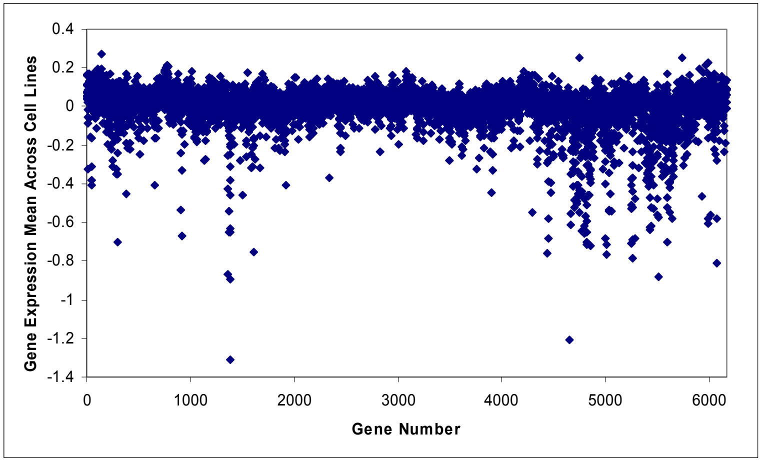


Figure 4.21: Mean Across Cell Lines for Ross Data

There appear to be several distinct groups of genes based on the mean gene expression levels shown in Figure 4.21. Some genes have means that could be considered outliers. Ideally, these observations should be examined in the lab to see if they are plausible. We have no reason to remove them here, as they might represent genes that are strongly active across all of the cell lines.

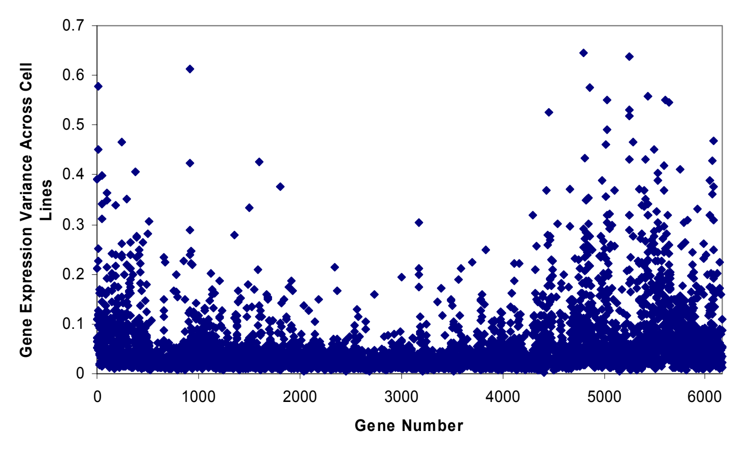


Figure 4.22: Variance Across Cell Lines for Ross Data

The gene expression variances shown in Figure 4.22 show a reasonable scatter. However, it is interesting that the variances for genes in the middle are lower than those for the genes on either end. This could be evidence of something systematic occurring. However, none of the variances have outrageous values and thus we proceed carefully.

The goal for this analysis is to find genes that simultaneously have small variances and small absolute values for the means. The plots in Figures 4.23 and 4.24 are helpful in selecting thresholds for such cases.

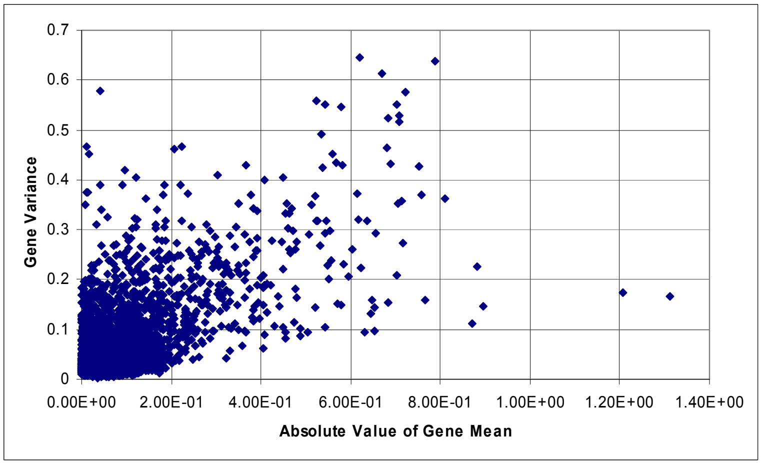


Figure 4.23: Absolute Gene Mean x Gene Variance Plot for Ross Data

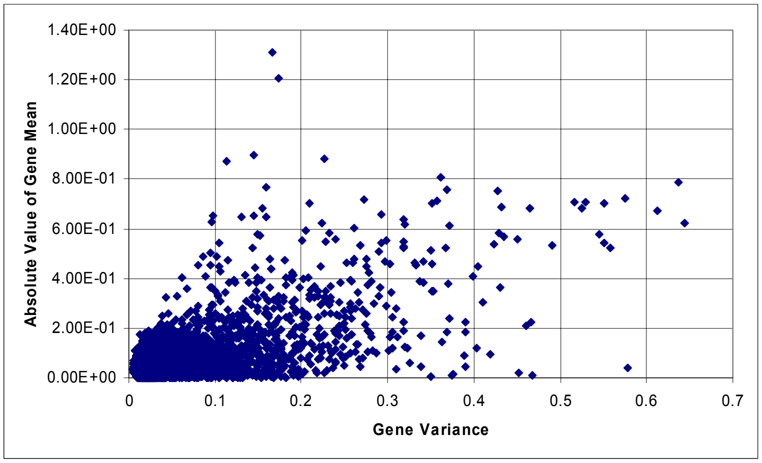


Figure 4.24: Gene Variance x Absolute Gene Mean Plot for Ross Data

It is desirable to eliminate a significant fraction of the noisy genes from further analysis. A large group of observations appear to have small variances and absolute means. By a somewhat arbitrary process, thresholds of 0.2 were chosen for both the absolute means and variances. Genes with absolute means and variances falling simultaneously below this value were filtered out. This resulted in 3,454 genes kept out of 6,165. This is a 44 percent reduction in the number of genes for the analysis.

The density for the 3,454 gene means is shown in Figure 4.25.

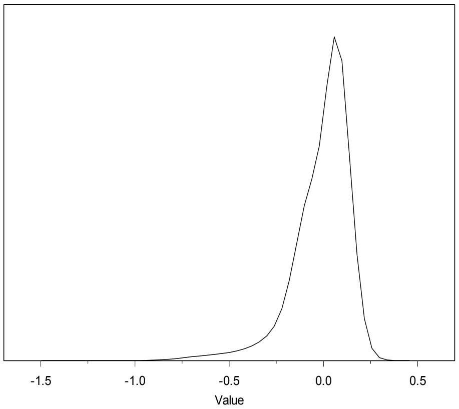


Figure 4.25: Density Plot for Ross Data

The density shown in Figure 4.19 has a long tail on the left side. This tail could make estimation more difficult. There are a few "bumps" in the curve which may represent clusters. One expects a rather large number of gene clusters to be present due to the large number of potential gene pathways. Normal mixture models were fit using various numbers of clusters. For each gene, the 60 cell lines were summarized using the mean. The model fit statistics are given in Table 4.16.

Table 4.16: Fit Statistics for Normal Mixture Model Applied to Genes for Ross Data

Number of Clusters	Log Likelihood	AIC	BIC
10	4057.50	-8057.01	-7878.74
11	4059.76	**-8055.52**	-7858.81
12	4082.22	-8094.43	-7879.28
13	4066.82	-8057.64	-7824.04
14	4098.38	-8114.76	-7862.72
15	4099.23	-8110.47	-7839.99
16	4105.86	-8117.72	-7828.79
17	4106.21	-8112.43	-7805.07
18	4105.82	-8105.64	-7779.83
19	4120.52	-8129.04	-7784.79
20	4109.70	-8101.40	-7738.71
21	4115.67	-8107.34	-7726.21
22	4124.41	-8118.82	-7719.25
23	4113.44	-8090.88	-7672.86
24	4118.66	-8095.33	-7658.87
25	4124.75	-8101.50	-7646.60
26	4126.26	-8098.52	-7625.18
27	4134.47	-8108.94	-7617.16
28	4142.69	-8119.38	-7609.16
29	4142.83	-8113.66	-7584.00
30	4148.36	-8118.72	-7571.61
35	4156.32	-8104.65	-7465.33
40	4181.51	-8125.02	-7393.50
45	Failed to converge		
50	**4206.14**	-8114.28	**-7198.33**

Figure 4.26 is a graphical representation of Table 4.16.

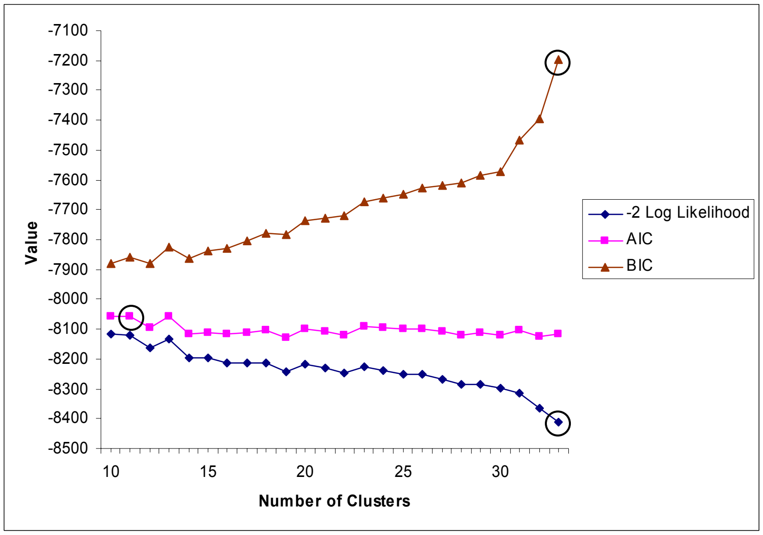


Figure 4.26: Plot of the -2 Log Likelihood, AIC, and BIC Values for Ross Data

The circled values in Figure 4.26 indicate the models selected by the three fit criteria. Since the BIC values appear to have an upward trend as the number of clusters increases, there are probably more clusters than the 50 present in the "best fitting" model. The idea that there are many clusters present for the genes is biologically sound, since genes may be involved in multiple pathways and there are many pathways. Due to space constraints, the 50 cluster model parameter estimates and cluster assignments are not given. In practice, one would continue fitting larger mixture models until satisfied that the BIC value no longer increases. This model would then be selected as the best fitting model.

Clustering a Subset of Genes Having Known Function

We compiled a gene list containing 429 genes known to be involved in specific genetic pathways. There are at least 21 pathways represented by these 429 genes. The means for the 429 genes are shown in the density plot in Figure 4.27.

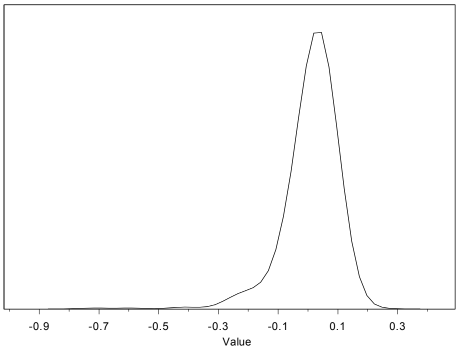


Figure 4.27: Density Plot for Ross Data

The mixture distribution looks complicated as shown in Figure 4.27. The distribution has a long left tail. There are several "bumps" present which may represent clusters. Normal mixture models of various cluster sizes were fitted. The fit statistics for these models are shown in Table 4.17.

Table 4.17: Fit Statistics for Normal Mixture Model Applied to Genes of Known Function for Ross Data

Number of Clusters	Log Likelihood	AIC	BIC
10	559.52	-1061.05	-943.27
11	562.79	-1061.59	-931.62
12	566.20	-1062.41	-920.26
13	569.63	-1063.27	-908.93
14	568.93	-1055.85	-889.33
15	571.15	-1054.29	-875.59
16	580.84	-1067.69	-876.80
17	573.60	-1047.21	-844.14
18	579.68	-1053.36	-838.10
19	*551.26*	*-1026.51*	*-872.18*
20	581.94	-1045.87	-806.25
21	*537.66*	*-991.32*	*-820.74*
22	*537.66*	***-987.32***	*-808.61*
23	*551.26*	*-1010.51*	*-823.69*
24	591.16	-1040.33	-751.97
25	599.31	-1050.62	-750.07
26	**605.89**	-1057.79	**-745.06**
27	*551.26*	*-994.51*	*-775.20*
28	*551.26*	*-990.51*	*-763.07*
29	*551.70*	*-987.40*	*-751.83*
30	*551.70*	*-983.40*	*-749.71*

The fit statistics given in Table 4.17 are plotted in Figure 4.28. The circles indicate the models selected by the three fit statistics. The bolded values in Table 4.17 indicate the models selected, while the italicized numbers indicate models that did not converge unless the cluster variances were forced to be equal. Notice that both the likelihood and the BIC suggest that the 26 cluster model fits best.

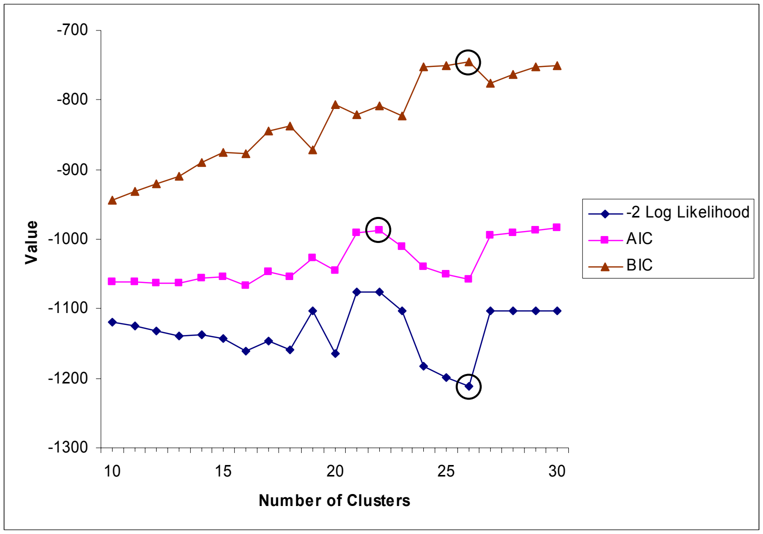


Figure 4.28: Plot of the -2 Log Likelihood, AIC, and BIC Values for Ross Data for Selected Genes

The 26 clusters were manually examined to see if any of the 429 genes clustered into the 21 known pathways. No clusters were found for which this was the case. These pathways would perhaps cluster better if the repeated measures mixture model discussed in Chapter 6 were applied. Collapsing the cell line observations for each gene using the mean may have removed much of the information that could have helped in grouping genes according to the pathways. The genes in a given pathway may not always cluster together under the best of circumstances.

4.4 Conclusion

The advantage of using parametric models, such as the normal mixture model, for clustering is that one can formally evaluate the model fit by using likelihood based statistics. Normal mixture models for clustering require the number of clusters to be specified. Section 4.3.5 introduces three criteria for evaluating the fit of a mixture model and suggests the BIC as the measure of choice. Alternative methods for evaluating the model fit are described in McLachlan and Peel (2000). Currently, multiple models having different numbers of clusters must be fit and the best fitting model selected using statistics such as the BIC. In order to lessen this workload, more research is needed to determine *a priori* how many clusters are expected. The gap statistic proposed by Tibshirani *et al.* (2000) is one such approach. However, the gap statistic requires multiple models to be fit before predicting the appropriate number of clusters. Selecting the best number of clusters is listed as a future research problem in Section 6.2.

In cases where large numbers of clusters are present, there are frequently a handful of very large clusters present. This circumstance could negatively effect the accuracy of the estimation for the numerous remaining smaller clusters. The large clusters often contain a high degree of noise. One option for analyzing such data is to remove the observations which are members of large clusters and to re-cluster the remaining observations. This technique may help in focusing on signals of interest which are often contained in small clusters.

When working with large numbers of observations, it is often helpful to filter the data in some way. Chapter 2 describes filtering in more detail. The filtering method applied in this chapter is a simple screening process based on means and variances both having to be below a threshold in order to be excluded. Filtering can speed up convergence time as well as result in more accurate parameter estimation.

The analyses of the two dimensional microarray data in this chapter required the data to be collapsed in one dimension and examined unidimensionally. These analyses resulted in clusters of cell lines or clusters of genes. However, some information may have been lost regarding the interrelationships between genes and cell lines. It may be helpful to examine the data with a truly two dimensional method, which would allow genes and cell lines to be simultaneously clustered. Such a method is developed in Chapter 5.

Chapter 5

Two Dimensional Parametric Clustering

5.1 Introduction

5.1.1 Motivation Behind Two Dimensional Parametric Clustering

Consider a data set that has 10 variables for 100 individuals. The one dimensional parametric model presented in Chapter 4 allows the researcher to group similar variables or similar individuals together, but not both simultaneously. The one dimensional model requires the data to be collapsed in some way across the dimension that is not grouped, since the data must be examined in one dimension at a time. This data reduction may mask interesting relationships between the data in the reduced dimension and the data in the dimension grouped across. Analyzing two dimensional data using one dimensional techniques requires the data to be first grouped in one dimension. The second dimension of the data can be scrutinized in a later subanalysis which regroups the initial groups in the other dimension. Such an approach is not optimal because of the potential loss of information due to clustering in stages.

Suppose that the researcher's question regards the interrelationships between the variables and the individuals. Such a question could be approached by extending the model developed in Chapter 4 to allow a two dimensional grouping of variables and individuals. Figure 5.1 gives an example of such two dimensional groupings.

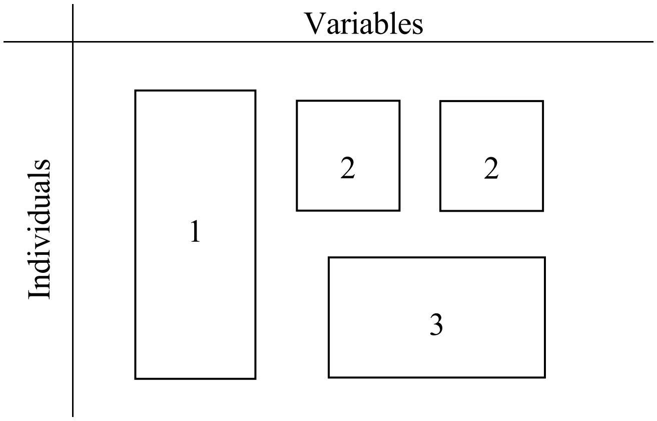


Figure 5.1: Two Dimensional Groups of Individuals and Variables

The boxes in the grid represent two dimensional groups labeled for easy identification. Groups 1, 2, and 3 demonstrate that the groups can take on different shapes. Group 1 is a group containing a few variables and many individuals. Group 2 is a disjoint group. Group 3 is a group containing a few individuals and many variables. The two dimensional model allows both disjoint groups and groups of varying shapes. Identifying which group a data point lies in requires two coordinates to be specified – the individual number and the variable number. This two dimensional relationship makes maximal use of the data. One dimensional techniques do not allow for this robust interpretation, since the data must be collapsed.

Two dimensional data are present in a variety of problems. Data are two dimensional when they can be represented in a grid with categorical labels across both axes. Some microarray data have a natural two dimensional format. One example is when the horizontal dimension represents *C* cell lines and the vertical dimension represents *G* genes. The observations are the gene expression values measured in fluorescence units. A model may be developed that allows the simultaneous grouping of cell lines and genes. Such a result may offer researchers new insight into groups of genes that are active in a set of cell types. It should be noted that this differs from a bivariate clustering where one may have data on two variables which are clustered simultaneously.

A popular approach for visualizing cluster results is a color map such as the one shown in Figure 5.2.

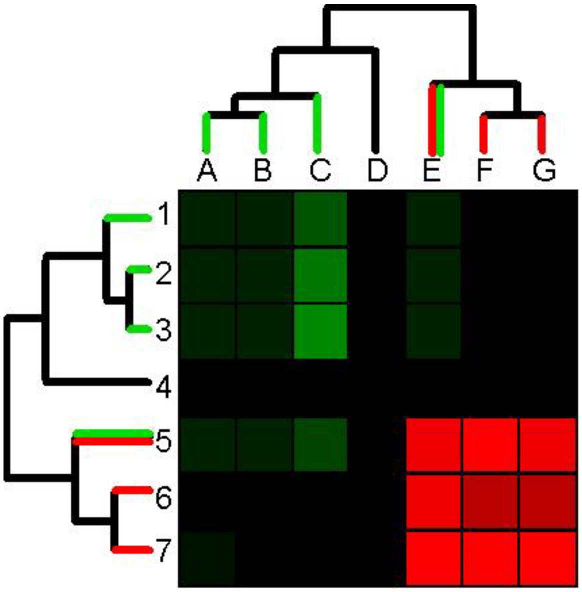


Figure 5.2: Example Color Map

The y axis of the color map represents one dimension of the data (genes), while the x axis represents the second dimension of the data (cell lines). A popular software package for constructing such color maps was written by Michael Eisen (1998). Although this color map looks two dimensional, it is actually the result of two one dimensional cluster analyses. A hierarchical clustering method is first applied to one dimension of the data. The dendrogram is positioned on the axis and the observations are shaded according to what clusters they fall in. The second dimension of the data is then clustered and the rows (or columns) are reordered such that observations falling in similarly sized clusters in both dimensions appear together. The second dendrogram is added to the plot. Such a representation of the data is helpful in visualizing cluster results. However, the analyses applied are not two dimensional, although they are often misinterpreted as such.

5.1.2 Chapter Contents

This chapter focuses on the development of two dimensional parametric clustering algorithms which cluster based on both dimensions of the data simultaneously. The motivation behind the development of two dimensional clustering models is discussed. Two dimensional data are described. The mixture model theory for the two dimensional case is introduced. The two dimensional normal mixture model is formulated. The Expectation/Maximization (EM) algorithm is outlined as an iterative likelihood technique that is useful for obtaining parameter estimates for the two dimensional mixture model. The appropriate extensions of the theory for implementing the EM algorithm are developed. The hybrid algorithm suggested for the one dimensional case (Hasselblad, 1966) is extended for the two dimensional mixture model case. Confidence interval formulas are derived for the estimates of the parameters and the posterior probabilities of a given observation belonging to a specific cluster. A software package, 2-DCluster, that implements these methods and calculates the appropriate confidence intervals is provided (see the 2-DCluster Appendix for the software documentation and availability). Data are simulated and analyzed. Experimental data from the Ross *et al.* (2000) microarray experiment are analyzed, followed by a discussion of the results.

5.2 The Two Dimensional Normal Mixture Model

5.2.1 Introduction to the Model

For performing two dimensional clustering, we suggest a model which directly extends the one dimensional normal mixture model discussed in Chapter 4. The two dimensional normal mixture model proposed could be written as:

$$f_{y_{ij}} = \sum_{k=1}^{C}\sum_{l=1}^{C^*} \pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^2\right), \tag{5.1}$$

where $\sigma_{kl}^2 = \sigma_k^2 + \sigma_{\tau l}^2$. For the two dimensional normal mixture model, the observations, y_{ij}, are associated with a row, indexed by *i,* and a column, indexed by *j*. This is illustrated in Table 5.1.

Table 5.1: Two Dimensional Normal Mixture Data

	j index
i index	y_{ij} (Data)

A single observation is represented by a pair of indices. The mixture model (introduced in Chapter 4) is formulated in terms of the random variable y_{ij}. One way to motivate the distribution given in Equation 5.1 is through a mixture of mixtures. Suppose, as in the one dimensional case, that the y_{ij} are distributed as mixtures of normal distributions $\phi\left(y_{ij};\ \mu_{kj},\sigma_k^2\right)$, where $i = 1,\ldots,N$ represents the *N* rows of data, $j = 1,\ldots,N^*$ represents the N^* columns of data, and $k = 1,\ldots,C$ represents the *C* groupings of the rows of data. Notice that the mean of the distribution of the k^{th} cluster, μ_{kj}, changes across the columns. (If the columns were treated as replicates this would reduce to the unidimensional case.) To accommodate clusters across the columns (*j*'s), the μ_{kj}'s themselves are assumed to be distributed as mixtures of normal distributions $\phi\left(\mu_{kj};\ \tau_{kl}, \sigma_{kl}^2\right)$, where *i*, *j*, and *k* are defined as before and $l = 1, \ldots, C^*$ represents the C^* groupings of the columns of the data. The normal mixture distribution of the Y_{ij} conditional on the fixed μ_{kj} is given by:

$$f = \sum_{k=1}^{C} \pi_k f_{x_{ij}|\mu_{kj}} = \sum_{k=1}^{C} \pi_k \phi\left(y_{ij};\ \mu_{kj}, \sigma_k^2\right). \tag{5.2}$$

The distributions of the μ_{kj}'s are assumed to be normal mixtures with the density:

$$g = \sum_{l=1}^{C^*} \psi_l f_{\mu_{kj}} = \sum_{l=1}^{C^*} \psi_l \phi\left(\mu_{kj};\ \tau_{kl}, \sigma_{\tau l}^2\right), \tag{5.3}$$

where ψ_l represents the proportion of observations falling in the l^{th} column cluster. Then the marginal distribution of y_{ij} is (see Appendix 5.1 for derivation details):

$$\begin{aligned} f_{y_{ij}} &= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_k\psi_l\int_{-\infty}^{\infty} f_{x_{ij}|\mu_{kj}} f_{\mu_{kj}}\partial\mu_{kj} \\ &= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_k\psi_l\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_k^2+\sigma_{\tau l}^2\right) \qquad (5.4) \\ &= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^2\right), \text{ where } \pi_{\mathrm{kl}}=\pi_k\psi_l \text{ and } \sigma_{kl}^2=\sigma_k^2+\sigma_{\tau l}^2. \end{aligned}$$

These types of marginal distributions are also called mixtures (McLachlan and Basford, 1988).

In this formulation, the one dimensional cluster specific parameters σ_k^2 and $\sigma_{\tau l}^2$ are not identifiable and only their sum, $\sigma_k^2+\sigma_{\tau l}^2$, is estimable. This is acceptable since the real parameters of interest are the two dimensional cluster specific parameters π_{kl}, τ_{kl}, and σ_{kl}^2. Notice that this reduces to a one dimensional form as follows:

$$f_{y_{ij}} = \sum_{m=1}^{CC^*}\pi_m\phi\left(y_{ij};\ \tau_{\mathrm{m}},\sigma_m^2\right), \qquad (5.5)$$

where $\pi_{\mathrm{m}}=\pi_{kl}$, $\sigma_m^2=\sigma_{kl}^2$, and the sum is over CC^* clusters. Such a simplification allows the formulas derived in Chapter 4 to be readily extended for the two dimensional case. When collapsed to vector notation, the two dimensional clustering framework preserves the two dimensional nature of the data and the clusters. Thus the problem reduces to maintaining the appropriate indices of the clusters corresponding to each y_{ij}.

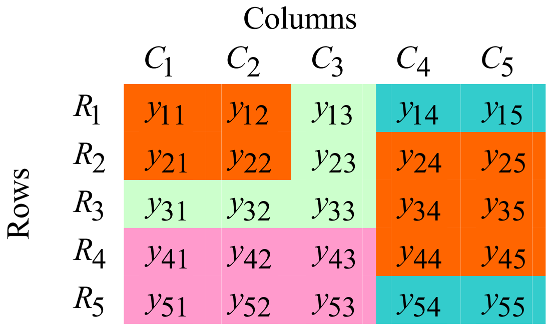
The expected value of y_{ij} is a weighted average,

$$E\left[y_{ij}\right]=\sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_{kl}\tau_{kl}. \tag{5.6}$$

The variance of y_{ij} is:

$$V\left[y_{ij}\right]=\sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_{kl}\left(\tau_{kl}^2+\sigma_{kl}^2\right)-\left[\sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_{kl}\tau_{kl}\right]^2. \tag{5.7}$$

The formulation of the two dimensional clustering framework (conditioning on rows or columns) is illustrated through the use of an example. Figure 5.3 shows a 5 x 5 grid of observations which come from a 4 cluster model.



			Columns		
Rows	C_1	C_2	C_3	C_4	C_5
R_1	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
R_2	y_{21}	y_{22}	y_{23}	y_{24}	y_{25}
R_3	y_{31}	y_{32}	y_{33}	y_{34}	y_{35}
R_4	y_{41}	y_{42}	y_{43}	y_{44}	y_{45}
R_5	y_{51}	y_{52}	y_{53}	y_{54}	y_{55}

Figure 5.3: Four Component Normal Mixture Example with 25 Observations

The different clusters shown in Figure 5.3 are described below. First define the following sets based on the four colors.

Red: $A=\left\{R_1,R_2\,|\,C_1\right\}$, $B=\left\{R_1,R_2\,|\,C_2\right\}$, $C=\left\{R_2,R_3,R_4\,|\,C_4\right\}$, $D=\left\{R_2,R_3,R_4\,|\,C_5\right\}$

Green: $E=\left\{R_3\,|\,C_1\right\}$, $F=\left\{R_3\,|\,C_2\right\}$, $G=\left\{R_1,R_2,R_3\,|\,C_3\right\}$

Pink: $H = \{R_4, R_5 \mid C_1\}$, $I = \{R_4, R_5 \mid C_2\}$, $J = \{R_4, R_5 \mid C_3\}$

Blue: $K = \{R_1, R_5 \mid C_4\}$, $L = \{R_1, R_5 \mid C_5\}$

The set defined by $A \cup B \cup C \cup D$ is shown in Equation 5.8.

$$\begin{aligned} CLUST1 &= \{\{R_1, R_2 \mid C_1\}, \{R_1, R_2 \mid C_2\}, \{R_2, R_3, y_4 \mid C_4\}, \{R_2, R_3, y_4 \mid C_5\}\} \\ &= \{y_{11}, y_{21}, y_{12}, y_{22}, y_{24}, y_{34}, y_{44}, y_{25}, y_{35}, y_{45}\} \end{aligned} \tag{5.8}$$

All of the observations contained in CLUST1 are members of the first cluster shown in red in Figure 5.3 and come from the normal distribution $\phi\left(y_{ij};\ \tau_{11}, \sigma_{11}^2\right)$. Since 10 out of 25 observations belong to cluster one, $\pi_{11} = 10/25 = 0.40$.

The set defined by $E \cup F \cup G$ is shown in Equation 5.9.

$$\begin{aligned} CLUST2 &= \{\{R_3 \mid C_1\}, \{R_3 \mid C_2\}, \{R_1, R_2, R_3 \mid C_3\}\} \\ &= \{y_{31}, y_{32}, y_{13}, y_{23}, y_{33}\} \end{aligned} \tag{5.9}$$

All of the observations contained in CLUST2 are members of the second cluster shown in green in Figure 5.3 and come from the normal distribution $\phi\left(y_{ij};\ \tau_{12}, \sigma_{12}^2\right)$. Since 5 out of 25 observations belong to cluster two, $\pi_{12} = 5/25 = 0.20$.

The set defined by $H \cup I \cup J$ is shown in Equation 5.10.

$$\begin{aligned} CLUST3 &= \{\{R_4, R_5 \mid C_1\}, \{R_4, R_5 \mid C_2\}, \{R_4, R_5 \mid C_3\}\} \\ &= \{y_{41}, y_{51}, y_{42}, y_{52}, y_{43}, y_{53}\} \end{aligned} \tag{5.10}$$

All of the observations contained in CLUST3 are members of the third cluster shown in pink in Figure 5.3 and come from the normal distribution $\phi\left(y_{ij};\ \tau_{21},\sigma_{21}^2\right)$. Since 6 out of 25 observations belong to cluster three, $\pi_{21} = 6/25 = 0.24$.

The set defined by $K \cup L$ is shown in Equation 5.11.

$$\begin{aligned} CLUST4 &= \left\{\left\{R_1, R_5 \mid C_4\right\}, \left\{R_1, R_5 \mid C_5\right\}\right\} \\ &= \left\{y_{24}, y_{34}, y_{44}, y_{25}, y_{35}, y_{45}\right\} \end{aligned} \tag{5.11}$$

All of the observations contained in CLUST4 are members of the fourth cluster shown in blue in Figure 5.3 and come from the normal distribution $\phi\left(y_{ij};\ \tau_{22},\sigma_{22}^2\right)$. Since 4 out of 25 observations belong to cluster four, $\pi_{22} = 4/25 = 0.16$.

The mixture distribution is formed from all of the observations. Each distribution is weighted by the proportion of observations that come from that distribution. The mixture distribution for this 4 component normal mixture is:

$$\begin{aligned} f_{y_{ij}} = &0.40\phi\left(y_{ij};\ \tau_{11},\sigma_{11}^2\right) + 0.20\phi\left(y_{ij};\ \tau_{12},\sigma_{12}^2\right) + 0.24\phi\left(y_{ij};\ \tau_{21},\sigma_{21}^2\right) + \\ &0.16\phi\left(y_{ij};\ \tau_{22},\sigma_{22}^2\right) \end{aligned} \tag{5.12}$$

The likelihood is:

$$\begin{aligned} L = \prod_{i=1}^{5}\prod_{j=1}^{5}\Big[&0.40\phi\left(y_{ij};\ \tau_{11},\sigma_{11}^2\right) + 0.20\phi\left(y_{ij};\ \tau_{12},\sigma_{12}^2\right) + \\ &0.24\phi\left(y_{ij};\ \tau_{21},\sigma_{21}^2\right) + 0.16\phi\left(y_{ij};\ \tau_{22},\sigma_{22}^2\right)\Big]. \end{aligned} \tag{5.13}$$

5.2.2 Parameter Estimation for Two Dimensional Normal Mixture Models

Suppose that a random sample of T observations is obtained from the two dimensional normal mixture described in Section 5.2.1, where N is the number of rows of the data, N^* is the number of columns of the data, and $T = N \bullet N^*$. The likelihood for the T observations is given by the joint density of the sample:

$$L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right) = \prod_{i=1}^{N} \prod_{j=1}^{N^*} \sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl} \phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^2\right), \tag{5.14}$$

where $\mathbf{y}$ is an $N \times N^*$ dimensional data matrix, $\boldsymbol{\pi}$ is a $CC^* - 1$ dimensional matrix of proportions, and $\boldsymbol{\tau}$ and $\boldsymbol{\sigma}^{\mathbf{2}}$ are CC^* dimensional vectors of means and variances, respectively. To obtain the maximum likelihood estimates (MLE's), the log likelihood,

$$\ell\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right) = \log\left[L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)\right] = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \log\left[\sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl} \phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^2\right)\right] \tag{5.15}$$

is maximized with respect to the parameters π_{kl}, τ_{kl}, and σ_{kl}^2. The MLE's are found by taking the first derivatives of the log likelihood with respect to the parameters of interest, setting them equal to zero, and solving.

Prior to solving the maximum likelihood equations, it is necessary to place a restriction on the proportions in order to insure that they sum to one. Two possible restrictions are the marginal and the global restrictions. These two restrictions lead to different expected values of *Y*, as shown below.

First, consider the mixture of mixtures case as defined in Equation 5.1. In this case, both π_k and ψ_l must sum to 1, which in terms of π_{kl} leads to $\pi_{kl} = \pi_k \psi_l$ and $\pi_{Cl} = 1 - \sum_{k=1}^{C-1} \pi_{kl}$ and $\pi_{kC^*} = 1 - \sum_{l=1}^{C^*-1} \pi_{kl}$. These restrictions force the marginals to conform to $\sum_{k=1}^{C} \pi_{k.} = 1$ and $\sum_{l=1}^{C^*} \pi_{.l} = 1$. The expected value of *Y* is calculated under these restrictions for the 2 x 2 normal mixture distribution. The π_{kl}'s for the 2 x 2 case are shown in Table 5.2, where $\pi_{kl} = \pi_{k.}\pi_{.l}$.

Table 5.2: Cluster Proportions for the 2 x 2 Case with Marginal Restrictions

π_{11}	π_{12}	$\pi_{1.}$
π_{21}	π_{22}	$\pi_{2.}$
$\pi_{.1}$	$\pi_{.2}$	1

The expected value of *Y* for the 2 x 2 case is:

$$\begin{aligned} E\left[y_{ij}\right] &= \sum_{k=1}^{C}\sum_{l=1}^{C^*} \pi_{kl}\tau_{kl} \\ &= \pi_{11}\tau_{11} + \pi_{12}\tau_{12} + \pi_{21}\tau_{21} + \pi_{22}\tau_{22} \\ &= \pi_{11}\tau_{11} + \left(\pi_{1.} - \pi_{11}\right)\tau_{12} + \pi_{21}\tau_{21} + \left(\pi_{2.} - \pi_{21}\right)\tau_{22} \\ &= \pi_{11}\left(\tau_{11} - \tau_{12}\right) + \pi_{1.}\tau_{12} + \pi_{21}\left(\tau_{21} - \tau_{22}\right) + \pi_{2.}\tau_{22} \\ &= \left(\pi_{1.}\pi_{.1}\right)\left(\tau_{11} - \tau_{12}\right) + \pi_{1.}\tau_{12} + \left(\pi_{2.}\pi_{.1}\right)\left(\tau_{21} - \tau_{22}\right) + \pi_{2.}\tau_{22} \\ &= \pi_{1.}\left[\pi_{.1}\left(\tau_{11} - \tau_{12}\right) + \tau_{12}\right] + \pi_{2.}\left[\pi_{.1}\left(\tau_{21} - \tau_{22}\right) + \tau_{22}\right]. \end{aligned} \tag{5.16}$$

Now consider the global restriction $\pi_{CC^*} = 1 - \sum_{\substack{k=1 \\ kl \neq CC^*}}^{C} \sum_{l=1}^{C^*} \pi_{kl}$. Table 5.3 shows the π_{kl} 's for the 2 x 2 case.

Table 5.3: Cluster Proportions for the 2 x 2 Case with Global Restrictions

π_{11}	π_{12}	
π_{21}	π_{22}	
		1

After applying the restriction, Table 5.3 may be rewritten as shown in Table 5.4.

Table 5.4: Cluster Proportions for the 2 x 2 Case After Global Restriction Application

π_{11}	$1-(\pi_{11}+\pi_{21}+\pi_{22})$
$1-(\pi_{11}+\pi_{12}+\pi_{22})$	$1-(\pi_{11}+\pi_{12}+\pi_{21})$

The expected value of *Y* is recalculated under this restriction for the 2 x 2 case.

$$\begin{aligned}
E\left[y_{ij}\right] &= \sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl}\tau_{kl} \\
&= \pi_{11}\tau_{11} + \pi_{12}\tau_{12} + \pi_{21}\tau_{21} + \pi_{22}\tau_{22} \\
&= \pi_{11}\tau_{11} + (1-\pi_{11}-\pi_{21}-\pi_{22})\tau_{12} + (1-\pi_{11}-\pi_{12}-\pi_{22})\tau_{21} + \\
&\quad (1-\pi_{11}-\pi_{12}-\pi_{21})\tau_{21} \qquad (5.17) \\
&= (\pi_{12}+\pi_{21}+\pi_{22}) + \pi_{11}(\tau_{11}-\tau_{12}-\tau_{21}-\tau_{22}) - \pi_{12}(\tau_{21}+\tau_{22}) - \\
&\quad \pi_{21}(\tau_{12}+\tau_{22}) - \pi_{22}(\tau_{12}+\tau_{21}) \\
&= (\pi_{12}+\pi_{2.}) + \pi_{11}(\tau_{11}-\tau_{12}-\tau_{21}-\tau_{22}) - \pi_{12}(\tau_{21}+\tau_{22}) - \\
&\quad \pi_{21}(\tau_{12}+\tau_{22}) - \pi_{22}(\tau_{12}+\tau_{21})
\end{aligned}$$

Notice that the expected values of Y in Equations 5.16 and 5.17 have different forms for the two sets of restrictions. Placing restrictions on the marginals and requiring the rows and the columns to be independent ($\pi_{kl} = \pi_k \pi_l$) causes the two dimensional clusters to be related through the margins and does not allow the flexibility of the clusters being able to take on any shape. This limitation is shown in Figure 5.4.

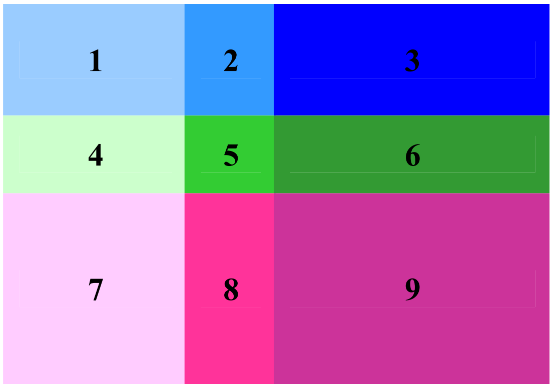


Figure 5.4: Two Dimensional Clusters with Marginal Restrictions

Observe in Figure 5.4 that the row and the column widths are fixed. This happens because the marginal restrictions fix the row or column widths for the clusters in one dimension and the two dimensional clusters must be formed from these fixed width component clusters. The global restriction on the π_{kl}'s does not force the clusters to lie in any particular position, as shown in Figure 5.5.

This flexibility arises from each observation being able to be a member of any cluster. Information regarding the observation's location in the data matrix is preserved by recording the row and column number from which it came from. This is useful in the two dimensional cluster application, so the global restriction $\pi_{CC^*} = 1 - \sum_{\substack{k=1 \\ kl \neq CC^*}}^{C} \sum_{l=1}^{C^*} \pi_{kl}$ is used in this dissertation.

The derivatives are calculated under this constraint.

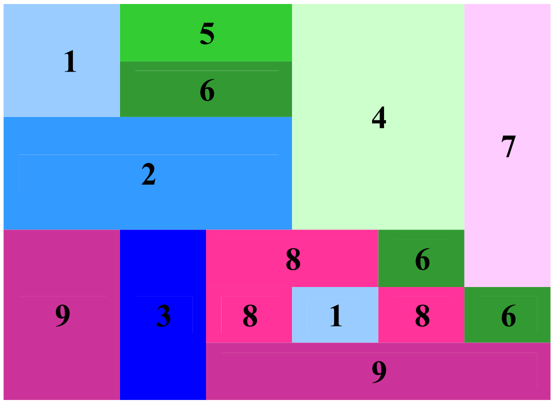


Figure 5.5: Two Dimensional Clusters with Global Restriction

Since the two dimensional clusters may take on any size or shape when the global constraint is applied, it is usually not possible to reorder the data so that observations belonging to the same cluster are adjacent to each other. This is due to the complex geometry of the clusters. Reordering the data may violate the cluster assignments generated by the algorithm. The inability to reorder the data eliminates the need to maintain two indices (*k* and *l*) for the clusters, as there is no way to separate the clusters that are due to the columns of the data from the clusters that are due to the rows of the data. This allows the substitution of a single summation over the CC^* clusters instead of the double summations over C and C^*. In other words, $\sum_{kl=1}^{CC^*}$ must be substituted for $\sum_{k=1}^{C} \sum_{l=1}^{C^*}$ in all of the formulas.

In some applications, the marginal constraints may be more appropriate. For example, if the researcher has prior knowledge of how one dimension of the data clusters, this information could be incorporated using the marginal constraints. Another application in which marginal constraints may be preferable is if an experimenter knows that the column and the row variables are independent. Applying the marginal constraints requires maintaining separate indices for the row and the column clusters. Thus, the double summation $\sum_{k=1}^{C}\sum_{l=1}^{C^*}$ must be included in all of the formulas for the marginal constraint case. This is because the fixed row or column cluster sizes, under these constraints, allow the data to be reordered in such a way that the observations belonging to the same cluster are adjacent to each other. As argued above, the global constraint seems more appropriate for microarray clustering applications, and thus is used in this dissertation.

Hasselblad (1966) calculates the first derivatives for the one dimensional normal mixture model. The derivatives for the two dimensional normal mixture model are shown below. Let the number of rows of the data, N, be indexed by i and the number of columns of the data, N^*, be indexed by j. Let C and C^* represent the number of groupings of the rows and columns of the data, respectively. Let y_{ij} be the $(i,j)^{th}$ observation, π_{kl} be the proportion of total observations contained in the $(k,l)^{th}$ cluster, and τ_{kl} and σ_{kl}^2 be the mean and variance, respectively, of the observations contained in the $(k,l)^{th}$ cluster. The distribution of the $(k,l)^{th}$ two dimensional normal mixture component is represented as:

$$f_{ijkl} = \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right) = \frac{1}{\sqrt{2\pi}\sigma_{kl}} \exp\left[-\frac{1}{2\sigma_{kl}^2}\left(y_{ij} - \tau_{kl}\right)^2\right]. \qquad (5.18)$$

The two dimensional normal mixture distribution of the random variable *Y* is written as:

$$F_{ij} = f_{y_{ij}}\left(y_{ij};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^2\right) = \sum_{k=1}^{C}\sum_{l=1}^{C^*} \pi_{kl}\phi\left(y_{ij};\ \tau_{kl},\ \sigma_{\mathrm{kl}}^2\right) = \sum_{\mathrm{k}=1}^{\mathrm{C}}\sum_{l=1}^{C^*} \pi_{\mathrm{kl}} f_{ijkl}. \qquad (5.19)$$

The first derivatives of the log likelihood are shown below.

$$\begin{aligned}
\frac{\partial \ell}{\partial \pi_{kl}} &= \sum_{i=1}^{N}\sum_{j=1}^{N^*} \frac{f_{ijkl} - f_{ijCC^*}}{F_{ij}}, \quad \mathrm{k} = 1, 2, \ldots, C \text{ and } \mathrm{l} = 1, 2, \ldots, C-1 \\
\frac{\partial \ell}{\partial \tau_{kl}} &= \sum_{i=1}^{N}\sum_{j=1}^{N^*} \frac{\pi_{kl} f_{ijkl}\left(y_{ij} - \tau_{kl}\right)}{\sigma_{kl}^2 F_{ij}}, \quad \mathrm{k} = 1, 2, \ldots, C \text{ and } \mathrm{l} = 1, 2, \ldots, C \qquad (5.20) \\
\frac{\partial \ell}{\partial \sigma_{kl}} &= \sum_{i=1}^{N}\sum_{j=1}^{N^*} \frac{\pi_{kl} f_{ijkl}}{F_{ij}}\left[\frac{\left(y_{ij} - \tau_{kl}\right)^2}{\sigma_{kl}^3} - \frac{1}{\sigma_{kl}}\right], \quad \mathrm{k} = 1, 2, \ldots, C \text{ and } \mathrm{l} = 1, 2, \ldots, C
\end{aligned}$$

These equations are nonlinear and therefore need to be solved iteratively. Due to the convergence problems with the Newton Raphson (NR) algorithm (Heath, 1997), the Expectation/Maximization (EM) algorithm is suggested. The EM and NR algorithms are introduced in Chapter 4.

To develop the EM algorithm, the first requirement is the definition of the incomplete data (Dempster *et al.*, 1977). For the two dimensional mixture model, the incomplete data are defined by the indicator variables that assign the observations to specific clusters. These indicator variables may be written as:

$$I_{ijkl} = \begin{cases} 1, \text{ if the } (i,j)^{th} \text{ observation } \in (k,l)^{th} \text{ cluster} \\ 0, \text{ otherwise} \end{cases} \tag{5.21}$$

The distributions of these indicator variables are specified by the posterior probabilities, α_{ijkl}. In other words, $P\left(I_{ijkl} = 1 \mid y_{ij}\right) = \alpha_{ijkl}$. In the Expectation stage of the EM algorithm, the complete data are obtained by estimating α_{ijkl}. The value α_{ijkl} represents the posterior probability of the $\left(i,j\right)^{th}$ observation falling in the $\left(k,l\right)^{th}$ cluster. The posterior probability is estimated by taking a weighted average of all CC^* of the posterior probabilities. That is,

$$E\left[I_{ijkl} \mid \mathbf{y}\right] = P\left[I_{ijkl} = 1 \mid \mathbf{y}\right] = \alpha_{ijkl} = \frac{\pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma^2_{\mathrm{kl}}\right)}{\sum_{r=1}^{C}\sum_{s=1}^{C^*} \pi_{kl}\ \phi\left(y_{ij};\ \tau_{rs},\ \sigma^2_{\mathrm{rs}}\right)}. \tag{5.22}$$

Notice that the number of unknowns in this step is $3CC^* - 1$. Thus, $3CC^* - 1$ initial values need to be specified: $CC^* - 1$ for the proportions, CC^* for the means, and CC^* for the variances. After the initial iteration, new estimates for these values are obtained from the maximization step of the EM algorithm.

The maximization step of the EM algorithm uses the completed data to estimate the parameters of the distribution. Once the observations falling into the different groups are identified, obtaining the $3CC^* - 1$ estimates of the proportions, means, and variances is straightforward and explicit solutions exist.

Equations 5.23 through 5.25 give the maximum likelihood estimates for the proportions, means, and variances for the normal mixture problem. For details of these derivations, see Appendix 5.2. The estimate for the proportion of observations falling in the $(k,l)^{th}$ cluster, π_{kl}, is:

$$\hat{\pi}_{kl} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl}}{NN^*} . \tag{5.23}$$

The estimate for the mean of the $(k,l)^{th}$ cluster, τ_{kl}, is:

$$\hat{\tau}_{kl} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N^*} y_{ij} \hat{\alpha}_{ijkl}}{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl}}. \tag{5.24}$$

The estimate for the variance of the $(k,l)^{th}$ cluster, σ^2_{kl}, is:

$$\hat{\sigma}^2_{kl} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl} \left(y_{ij} - \hat{\tau}_{kl} \right)^2}{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl}}. \tag{5.25}$$

For the case of homogenous variances among the CC^* clusters, the σ_{kl}^2's are first estimated for all of the clusters. Once these estimates are found, the new common variance, σ^2, is estimated by:

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{C}\sum_{l=1}^{C^*}\hat{\sigma}_{kl}^2\left(\sum_{i=1}^{N}\sum_{j=1}^{N^*}\hat{\alpha}_{ijkl}\right)}{NN^*} \quad . \tag{5.26}$$

Each iteration of the EM algorithm involves calculating equations 5.22 through 5.25 in that sequence. For the first iteration, the starting values are used to calculate $\hat{\alpha}_{ijkl}$. At the end of each iteration, a check is made to see if the EM algorithm converged. As described in Section 4.3.2, there are a number of ways that this check can be performed. The approach taken by McLachlan and Peel (2000) is used in this dissertation. A check for convergence is performed by examining all CC^* of the proportion (π_{kl}) estimates and seeing if the change from the previous iteration's estimate is less than some tolerance value. If any one of the CC^* estimates has changed by some amount greater than the tolerance value, the algorithm continues. The justification for this stopping rule is that the proportions are the most important parameters to estimate accurately because they indicate the relative sizes of the clusters to the user and contain the $\sum_{i=1}^{N}\sum_{j=1}^{N^*}\hat{\alpha}_{ijkl}$ terms which are involved in both the mean and the variance estimates.

Cluster membership is assigned by calculating all CC^* of the $\hat{\alpha}_{ijkl}$'s for each individual y_{ij} and assigning the individual to the cluster for which the posterior probability of belonging is greatest. However, for values of $\hat{\alpha}_{ijkl}$ that are very close to each other, assigning an individual to a cluster based on the maximum posterior probability may not be optimal. All of the posterior probabilities are available from the mixture model fit, and the user can evaluate the results of different cluster assignments. Section 4.3.5 describes approaches for comparing models that have different numbers of clusters.

5.2.3 The EM/Newton-Raphson Hybrid Algorithm

The EM/Newton-Raphson hybrid algorithm takes advantage of the best features of the Expectation/Maximization (EM) and Newton-Raphson (NR) algorithms in order to decrease the convergence time and to obtain an accurate solution. The implementation of the hybrid algorithm for the one dimensional normal mixture problem is described in Section 4.3.3. Implementing the algorithm for the two dimensional normal mixture problem requires extending the formulas to the two dimensional case. Otherwise, the implementation steps are the same.

Equations 5.1 – 5.26 extend the one dimensional normal mixture formulas for the EM algorithm to the two dimensional normal mixture case. Equation 5.20 gives the first derivatives of the log likelihood necessary for applying the NR algorithm. The NR algorithm also requires the second derivatives of the log likelihood, which are described below. First, the Kronecker delta is extended to the two dimensional case. Define δ_{mnkl} as the two dimensional Kronecker delta,

$$\delta_{mnkl} = \begin{cases} 1, \text{ if } m = k \text{ and } n = l \\ 0, \text{ otherwise} \end{cases}. \tag{5.27}$$

The second derivatives of the log likelihood contain δ_{mnkl} and are given below.

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \tau_{mn} \partial \tau_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{-\pi_{mn} \pi_{kl} f_{ijmn} f_{ijkl}}{\sigma_{mn}^2 \sigma_{kl}^2 F_{ij}^2} \left(y_{ij} - \tau_{mn} \right) \left(y_{ij} - \tau_{kl} \right) + \\ \delta_{mnkl} \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{\pi_{kl} f_{ijkl}}{F_{ij}} \left[\frac{\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^4} - \frac{1}{\sigma_{kl}^2} \right] \end{aligned} \tag{5.28}$$

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \sigma_{mn} \partial \tau_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{-\pi_{mn} \pi_{kl} f_{ijmn} f_{ijkl}}{\sigma_{kl}^2 F_{ij}^2} \left(y_{ij} - \tau_{kl} \right) \left[\frac{-1}{\sigma_{mn}} + \frac{\left(y_{ij} - \tau_{mn} \right)^2}{\sigma_{mn}^3} \right] + \\ \delta_{mnkl} \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{\pi_{kl} f_{ijkl}}{F_{ij}} \left(y_{ij} - \tau_{kl} \right) \left[\frac{-3}{\sigma_{kl}^3} + \frac{\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^5} \right] \end{aligned} \tag{5.29}$$

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \pi_{mn} \partial \tau_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{-\pi_{kl} f_{ijkl}}{\sigma_{kl}^2 F_{ij}^2} \left(y_{ij} - \tau_{kl} \right) \left(f_{ijmn} - f_{ijCC^*} \right) + \\ \left(\delta_{mnkl} - \delta_{klCC^*} \right) \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{f_{ijkl} \left(y_{ij} - \tau_{kl} \right)}{\sigma_{kl}^2 F_{ij}} \end{aligned} \tag{5.30}$$

$$\begin{aligned} \frac{\partial^2 \ell}{\partial \pi_{mn} \partial \sigma_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{-\pi_{kl} f_{ijkl}}{F_{ij}^2} \left(f_{ijmn} - f_{ijCC^*} \right) \left[\frac{\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^3} - \frac{1}{\sigma_{kl}} \right] + \\ \left(\delta_{mnkl} - \delta_{klCC^*} \right) \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{f_{ijkl}}{F_{ij}} \left[\frac{\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^3} - \frac{1}{\sigma_{kl}} \right] \end{aligned} \tag{5.31}$$

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \sigma_{mn} \partial \sigma_{kl}} = & \sum_{i=1}^{N} \sum_{j=1}^{N^*} \left(\left[\frac{-\pi_{mn} \pi_{kl} f_{ijmn} f_{ijkl}}{F_{ij}^2} \right] \left[\left\{ \frac{\left(y_{ij} - \tau_{mn} \right)^2}{\sigma_{mn}^3} - \frac{1}{\sigma_{mn}} \right\} \right. \right. \\ & \left. \left. x \left\{ \frac{\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^3} - \frac{1}{\sigma_{kl}} \right\} \right] \right) \\ & + \delta_{mnkl} \sum_{i=1}^{N} \sum_{j=1}^{N^*} \left(\left[\frac{\pi_{kl} f_{ijkl}}{F_{ij}} \right] \left[\frac{-1}{\sigma_{kl}} \right] \left[\frac{3\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^3} - \frac{1}{\sigma_{kl}} \right] \right. \\ & \left. x \left[\frac{\left(y_{ij} - \tau_{kl} \right)^2}{\sigma_{kl}^3} - \frac{1}{\sigma_{kl}} \right]^2 \right) \end{aligned} \qquad (5.32)$$

$$\frac{\partial^2 \ell}{\partial \pi_{mn} \partial \pi_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \frac{-\left(f_{ijkl} - f_{ijCC^*} \right) \left(f_{ijmn} - f_{ijCC^*} \right)}{F_{ij}^2} \qquad (5.33)$$

Equations 5.28 – 5.33 are used to form the Hessian matrix necessary for implementing the Newton-Raphson algorithm. See Section 4.3.3 for further details on the NR and EM/NR hybrid algorithms.

5.2.4 Calculating Confidence Intervals

In this section, confidence intervals are derived for the posterior probability of the $(i, j)^{th}$ observation falling in the $(k,l)^{th}$ cluster, or α_{ijkl}. The value of the α_{ijkl}'s determines which cluster an observation is assigned to. For values of α_{ijkl} that are close together for multiple (k,l) pairs, a confidence interval may help the user to determine which cluster to place the observation, y_{ij}, in. If the confidence intervals for α_{ijkl} overlap for a given (i, j) pair, the observation y_{ij} could be placed in more than one cluster. Thus, the confidence intervals for α_{ijkl} help to support the use of overlapping clusters. Overlapping clusters may be desirable in some clustering situations.

The derivation of the confidence interval for α_{ijkl} makes use of the delta method. The delta method is frequently used for finding the asymptotic variances of functions of parameters (Bickel and Doksum, 2001). Equation 5.34 is obtained by plugging the parameter estimates into Equation 5.22, or:

$$\hat{\alpha}_{ijkl} = \frac{\hat{\pi}_{kl}\ \phi\left(y_{ij};\ \hat{\tau}_{kl},\ \hat{\sigma}^2_{\mathrm{kl}}\right)}{\sum_{r=1}^{C}\sum_{s=1}^{C^*}\hat{\pi}_{rs}\ \phi\left(y_{ij};\ \hat{\tau}_{rs},\ \hat{\sigma}^2_{\mathrm{rs}}\right)}. \tag{5.34}$$

The delta method has the form:

$$V\left(\hat{\alpha}_{ijkl}\right) = f'\left(\underline{\theta}\right)\Sigma\left[f'\left(\underline{\theta}\right)\right]^T, \tag{5.35}$$

where $\underline{\theta}$ represents the vector of unknown parameters in $\hat{\alpha}_{ijkl}$, which are π_{kl}, τ_{kl}, and σ_{kl}. The *T* refers to the matrix transpose operation. The first derivative of *f* with respect to these parameters is:

$$f^{'}\left(\underline{\theta}\right)=\left[\frac{\partial\widehat{\alpha}_{ijkl}}{\partial\pi_{kl}}\;\frac{\partial\widehat{\alpha}_{ijkl}}{\partial\tau_{kl}}\;\frac{\partial\widehat{\alpha}_{ijkl}}{\partial\sigma_{kl}}\right]. \qquad (5.36)$$

The symmetric matrix Σ is:

$$\begin{bmatrix} Var\left(\pi_{kl}\right) & Cov(\tau_{kl},\pi_{kl}) & Cov(\sigma_{kl},\pi_{kl}) \\ Cov\left(\pi_{kl},\tau_{kl}\right) & Var(\tau_{kl}) & Cov(\sigma_{kl},\tau_{kl}) \\ Cov(\pi_{kl},\sigma_{kl)} & Cov(\tau_{kl},\sigma_{kl}) & Var(\sigma_{kl}) \end{bmatrix} \qquad (5.37)$$

The derivatives needed for Equation 5.35 are shown below and are derived in Appendix 5.3.

The derivative of $\widehat{\alpha}_{ijkl}$ with respect to π_{kl} is:

$$\frac{\partial\widehat{\alpha}_{ijkl}}{\partial\pi_{kl}}=\frac{\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)-\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\right]}{\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right]^{2}}. \qquad (5.38)$$

The derivative of $\widehat{\alpha}_{ijkl}$ with respect to τ_{kl} is:

$$\frac{\partial\widehat{\alpha}_{ijkl}}{\partial\tau_{kl}}=\left\{\pi_{kl}\left(y_{ij}-\tau_{kl}\right)\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\left(\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right.\right.$$
$$\left.\left.-\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\right)\right\}\ /\ \left\{\sigma_{kl}^{4}\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right]^{2}\right\}. \qquad (5.39)$$

The derivative of $\widehat{\alpha}_{ijkl}$ with respect to σ_{kl} is:

$$\frac{\partial \widehat{\alpha}_{ijkl}}{\partial \sigma_{kl}} = \left(\frac{\sum_{r=1}^{C} \sum_{s=1}^{C^*} \pi_{rs} \phi\left(y_{ij};\ \tau_{\text{rs}}, \sigma_{rs}^2 \right) - \pi_{kl} \left[\phi\left(y_{ij};\ \tau_{\text{rs}}, \sigma_{rs}^2 \right) \right]}{\left[\sum_{r=1}^{C} \sum_{s=1}^{C^*} \pi_{rs} \phi\left(y_{ij};\ \tau_{\text{rs}}, \sigma_{rs}^2 \right) \right]^2} \right). \tag{5.40}$$

Note that all of the parameter estimates are required to solve Equations 5.38 – 5.40. The elements of Σ are obtained from the corresponding elements of $I^{-1}\left(\underline{\theta}\right)$, where *I* is the Fisher information matrix. This is calculated by taking the negative of the inverse of the Hessian matrix defined in Chapter 4. The Hessian matrix does not include the estimate for π_{CC^*} since it can be found from the constraint $\pi_{CC^*} = 1 - \sum_{\substack{k=1 \\ kl \neq CC^*}}^{C} \sum_{l=1}^{C^*} \pi_{kl}$. The missing elements of the Hessian matrix related to π_{CC^*} are obtained by plugging the estimated values into the appropriate derivatives. Once the algorithm terminates, estimates for all of the parameter values are available.

A confidence interval for posterior probability, α_{ijkl}, may be calculated as:

$$\widehat{\alpha}_{ijkl} \pm z_{1-\alpha/2} \sqrt{Var(\widehat{\alpha}_{ijkl})}, \tag{5.41}$$

where $z_{1-\alpha/2}$ is the usual $100(1-\alpha/2)$ percentile of the standard normal distribution. For example, to obtain 95 percent confidence limits, $\alpha = 0.05$ and $z_{1-\alpha/2} = 1.96$.

Confidence intervals may also be calculated for the cluster specific parameters. Equation 5.42 gives the confidence interval formula for the group proportion, π_{kl}.

$$\hat{\pi}_{kl} \pm z_{1-\alpha/2}\sqrt{Var(\hat{\pi}_{kl})} \tag{5.42}$$

Equation 5.43 gives the confidence interval formula for the group mean, τ_{kl}.

$$\hat{\tau}_{kl} \pm z_{1-\alpha/2}\sqrt{Var(\hat{\tau}_{kl})} \tag{5.43}$$

Equation 5.44 gives the confidence interval formula for the group standard deviation, σ_{kl}.

$$\hat{\sigma}_{kl} \pm z_{1-\alpha/2}\sqrt{Var(\hat{\sigma}_{kl})} \tag{5.44}$$

By the invariance property of the MLE, the confidence intervals for the group variance, σ_{kl}^2, are found by squaring the estimate obtained for σ_{kl} and applying the usual confidence interval formula.

5.2.5 Analysis of Simulated Data

Simulated 4 Component Two Dimensional Normal Mixture

A four component two dimensional normal mixture model is simulated in order to introduce two dimensional clustering. For illustration purposes, three four component two dimensional normal mixture distributions are simulated and analyzed. The first simulation has different proportions, well separated means, and small variances. The second simulation has equal proportions and different means and variances. The third simulation has equal proportions and variances but different means. The SAS code for simulating two dimensional normal mixture distributions is given in Appendix 5.4.

The data for the first two dimensional mixture distribution are simulated using the parameter values shown in Table 5.5.

Table 5.5: Parameters for 4 Component Two Dimensional Normal Mixture Model with Different Proportions, Different Means, and Different Variances

Cluster Number	Proportion	Mean	Variance
1	0.600	5.000	1.000
2	0.200	10.000	2.000
3	0.100	15.000	3.000
4	0.100	20.000	4.000

Even for such well separated clusters, the normal mixture density is complicated as shown in Figure 5.6.

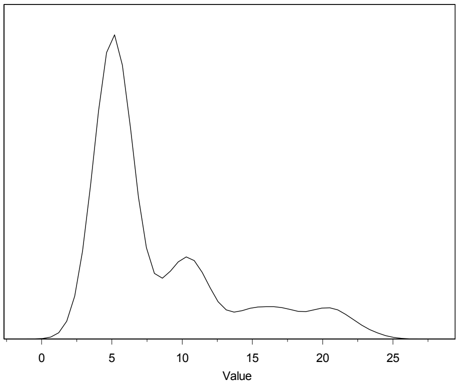


Figure 5.6: 4 Component Two Dimensional Normal Mixture Density Plot for First Simulation

There are several modes visible in Figure 5.6. The large mode on the left is due to a large percentage of the observations being concentrated in this region. The density is rather flat on the right hand side. This could make the estimation more difficult. There were 1,000 observations simulated according the parameters given in Table 5.5. Each observation was randomly generated from one of four normal distributions. Each observation was also randomly assigned a unique (row, column) pair. The rows assigned ranged from 1 – 10 and the columns assigned ranged from 1 – 100. The two dimensional parametric clustering algorithm differs from the one dimensional parametric clustering algorithm in that it keeps track of the two dimensional indices. A color map could be constructed from the (row, column) indices associated with each observation. Each of the four clusters would be represented in a different color. The cluster assignments are made based on the maximum posterior probability.

A 4 component two dimensional mixture distribution was fitted. The starting values were generated from a k-means cluster analysis. These starting values are shown in Table 5.6.

Table 5.6: Starting Values for First Simulation of 4 Component Two Dimensional Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.079	20.687	1.902
2	0.094	15.766	2.025
3	0.195	10.270	1.420
4	0.632	5.104	0.971

The model converged in 1 second on a 1.5 gigahertz Windows XP machine and required 165 EM algorithm iterations. The log likelihood for the model was -2504.63, the AIC was 5031.26, and the BIC was 5085.25. The parameter estimates are reported in Table 5.7.

Table 5.7: Fit Results for First Simulation of 4 Component Two Dimensional Normal Mixture Model

Actual Parameters			Estimated Parameters		
Proportion	Mean	Variance	Proportion	Mean	Variance
0.600	5.000	1.000	0.629	5.099	1.970
0.200	10.000	2.000	0.196	10.228	1.661
0.100	15.000	3.000	0.100	15.898	3.189
0.100	20.000	4.000	0.074	20.725	2.103

Notice in Table 5.7 that the proportions and means were estimated reasonably accurately. The estimated variances are not as accurate. In general, the mean estimation is more accurate than the variance estimation. This is due to the difficulty in discerning where the tails of the normal component distributions lie in the mixture distribution. The best case cluster labeling algorithm described in chapter 2 is applied. The kappa statistic is 0.906, the weighted kappa statistic is 0.943, the Rand index is 0.962, and the adjusted Rand index is 0.924. All of these statistics (discussed in Chapter 2) indicate good agreement between the mixture model clustering and the true simulated clusters.

The parameters for the second 4 component two dimensional normal mixture distribution simulation are given in Table 5.8.

Table 5.8: Parameters for 4 Component Two Dimensional Normal Mixture Model with Equal Proportions, Different Means, and Different Variances

Cluster Number	Proportion	Mean	Variance
1	0.250	5.000	1.000
2	0.250	10.000	2.000
3	0.250	15.000	3.000
4	0.250	20.000	4.000

Even for such well separated clusters, the normal mixture density is complicated as shown in Figure 5.7.

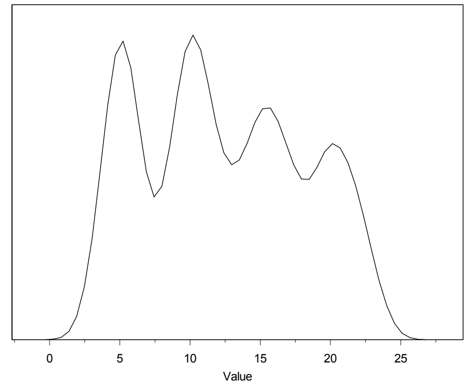


Figure 5.7: 4 Component Two Dimensional Normal Mixture Density Plot for Second Simulation

There are four distinct modes present in Figure 5.7. This is the ideal situation for the estimation of the component density parameters. There were 1,000 observations simulated according the parameters given in Table 5.8. Each observation was randomly generated from one of four normal distributions. Each observation was also randomly assigned a unique (row, column) pair. The rows assigned ranged from 1 – 10 and the columns assigned ranged from 1 – 100. The two dimensional parametric clustering algorithm differs from the one dimensional parametric clustering algorithm in that it keeps track of the two dimensional indices. A color map could be constructed from the (row, column) indices associated with each observation. Each of the four clusters would be represented in a different color. The cluster assignments are made based on the maximum posterior probability.

A 4 component two dimensional mixture distribution was fitted. The starting values were generated from a k-means cluster analysis. These starting values are shown in Table 5.9.

Table 5.9: Starting Values for Second Simulation of 4 Component Two Dimensional Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.216	20.547	2.159
2	0.236	15.436	1.630
3	0.292	10.369	1.526
4	0.256	5.167	1.161

The model converged in 1 second on a 1.5 gigahertz Windows XP machine and required 256 EM algorithm iterations. The log likelihood for the model was -2959.91, the AIC was 5941.83, and the BIC was 5995.81. The parameter estimates are reported in Table 5.10.

Table 5.10: Fit Results for Second Simulation of 4 Component Two Dimensional Normal Mixture Model

Actual Parameters			Estimated Parameters		
Proportion	Mean	Variance	Proportion	Mean	Variance
0.250	5.000	1.000	0.246	5.086	1.021
0.250	10.000	2.000	0.278	10.132	1.646
0.250	15.000	3.000	0.279	15.404	3.346
0.250	20.000	4.000	0.196	20.697	2.119

Notice in Table 5.10 that the proportions and the means were estimated reasonably accurately. The variance estimates are not as accurate. The best case cluster labeling algorithm described in chapter 2 is applied. The kappa statistic is 0.890, the weighted kappa statistic is 0.932, the Rand index is 0.926, and the adjusted Rand index is 0.805. All of these statistics (discussed in Chapter 2) indicate good agreement between the mixture model clustering and the true simulated clusters.

The parameters for the third 4 component two dimensional normal mixture distribution simulation are given in Table 5.11.

Table 5.11: Parameters for 4 Component Two Dimensional Normal Mixture Model with Equal Proportions, Different Means, and Equal Variances

Cluster Number	Proportion	Mean	Variance
1	0.250	5.000	2.000
2	0.250	10.000	2.000
3	0.250	15.000	2.000
4	0.250	20.000	2.000

Even for such well separated clusters, the normal mixture density is complicated as shown in Figure 5.8.

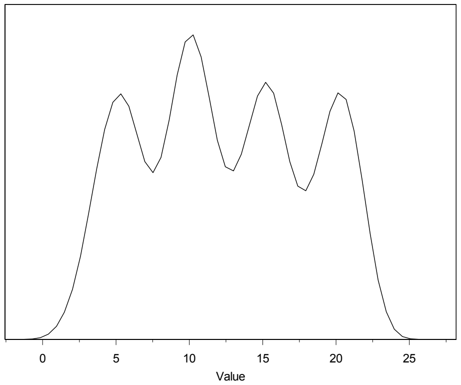


Figure 5.8: 4 Component Two Dimensional Normal Mixture Density Plot for Third Simulation

There are four distinct modes present in Figure 5.8. This is the ideal situation for the estimation of the component density parameters. There were 1,000 observations simulated according the parameters given in Table 5.11. Each observation was randomly generated from one of four normal distributions. Each observation was also randomly assigned a unique (row, column) pair. The rows assigned ranged from 1 – 10 and the columns assigned ranged from 1 – 100. The two dimensional parametric clustering algorithm differs from the one dimensional parametric clustering algorithm in that it keeps track of the two dimensional indices. A color map could be constructed from the (row, column) indices associated with each observation.

Each of the four clusters would be represented in a different color. The cluster assignments are made based on the maximum posterior probability.

A 4 component two dimensional mixture distribution was fitted. The starting values were generated from a k-means cluster analysis. These starting values are shown in Table 5.12.

Table 5.12: Starting Values for Third Simulation of 4 Component Two Dimensional Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.228	20.331	1.891
2	0.288	10.125	1.751
3	0.245	15.170	1.846
4	0.239	4.961	2.371

The model converged in 1 second on a 1.5 gigahertz Windows XP machine and required 23 EM algorithm iterations. The relatively short number of iterations required for convergence is due in part to the estimation of a common variance term. The log likelihood for the model was -3048.45, the AIC was 6112.89, and the BIC was 6152.16. The parameter estimates are reported in Table 5.13.

Table 5.13: Fit Results for Third Simulation of 4 Component Two Dimensional Normal Mixture Model

Actual Parameters			Estimated Parameters		
Proportion	Mean	Variance	Proportion	Mean	Variance
0.250	5.000	2.000	0.236	5.007	2.511
0.250	10.000	2.000	0.298	10.145	2.511
0.250	15.000	2.000	0.241	15.235	2.511
0.250	20.000	2.000	0.227	20.270	2.511

Notice in Table 5.13 that the proportions and means were estimated reasonably accurately. The estimated common variance is not as accurate. The best case cluster labeling algorithm described in chapter 2 is applied. The kappa statistic is 0.853, the weighted kappa statistic is 0.910, the Rand index is 0.899, and the adjusted Rand index is 0.731. All of these statistics (discussed in Chapter 2) indicate good agreement between the mixture model clustering and the true simulated clusters.

Simulated 20 Component Two Dimensional Normal Mixture

We simulated 100 sets of 1,000 observations for a 20 component two dimensional normal mixture. The same random number seeds were used throughout the simulation. The parameters used for the simulation are given in Table 5.14. A k-means cluster analysis was used to generate the starting values.

The means (Equation 5.6) and variances (Equation 5.7) of *Y* for the mixture distribution are comparable for multiple runs of simulated data. For the data simulated using the parameter values reported in Table 5.14, Figure 5.9 shows a scatter plot of the expected values of *Y*.

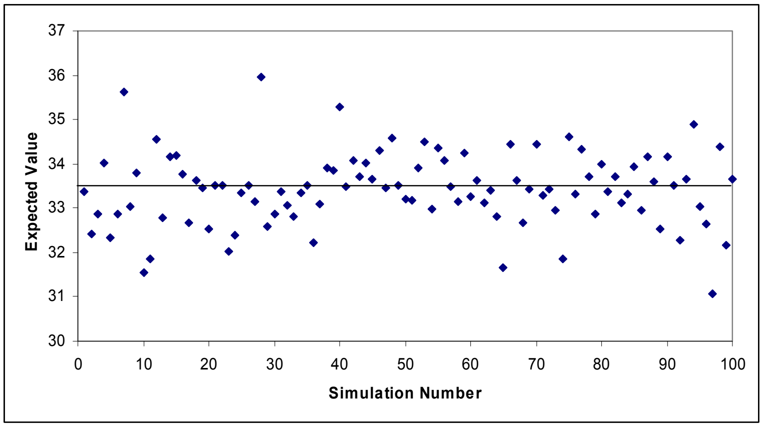


Figure 5.9: Scatter Plot of Expected Values of *Y* for 20 Component Two Dimensional Normal Mixture Simulation with 100 Runs

Table 5.14: Parameters for Simulation of 20 Component Two Dimensional Normal Mixture Model

Cluster Number	Proportion	Mean	Variance
1	0.240	5.000	1.000
2	0.120	10.000	2.000
3	0.060	15.000	3.000
4	0.090	20.000	4.000
5	0.090	25.000	5.000
6	0.040	30.000	6.000
7	0.020	35.000	7.000
8	0.010	40.000	8.000
9	0.015	45.000	9.000
10	0.016	50.000	10.000
11	0.060	55.000	11.000
12	0.030	60.000	12.000
13	0.015	65.000	13.000
14	0.022	70.000	14.000
15	0.022	75.000	15.000
16	0.060	80.000	16.000
17	0.030	85.000	17.000
18	0.015	90.000	18.000
19	0.022	95.000	19.000
20	0.022	100.000	20.000

The actual mean was 33.5 and is indicated by the horizontal line in Figure 5.9. The mean of the 100 simulations was 33.42 with a standard error of 0.83. Figure 5.10 shows a scatter plot of the variances for *Y*.

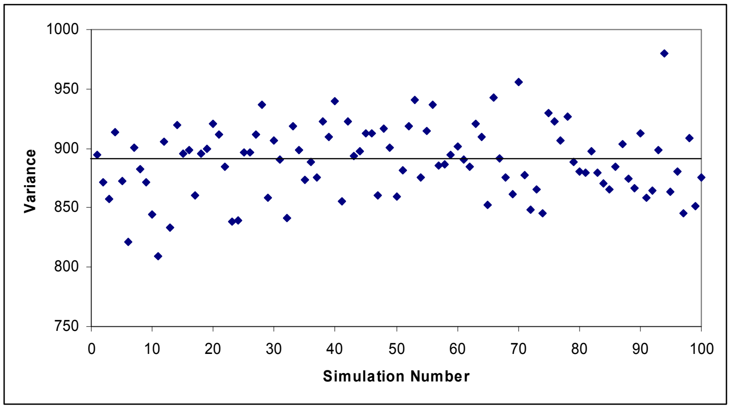


Figure 5.10: Scatter Plot of Variances of *Y* for 20 Component Two Dimensional Normal Mixture Simulation with 100 Runs

The actual variance was 892.57 and is indicated by the horizontal line in Figure 5.10. The variance for the 100 simulations was 889.10 with a standard error of 30.09. The estimation appears to be recovering the parameters reasonably well.

5.2.6 Analysis of the Ross *et al.* (2000) Data Set

The data set from Ross *et al.* (2000) was analyzed using two dimensional normal mixture model based clustering. The full data set has expression values for 6165 genes and 60 cell lines, for a total of 369,900 observations. Such a large data set is computationally difficult to analyze using mixture models even after reducing the size of the data set via filtering. Thus, for illustration purposes, we analyzed a subset of the Ross *et al.* (2000) data which is described below.

For the data set from Ross *et al.* (2000) (described in detail in Section 4.2), we compiled a gene list containing 429 genes known to be involved in specific genetic pathways. These genes are listed in Appendix 5.5. There are at least 21 pathways represented by these 429 genes. Each of these genes has fluorescence measurements for 60 cell lines. Thus, there are a total of 429 x 60 = 25,740 observations. For computational convenience and to remove non-informative observations that are likely due to noise, this data was filtered using the method suggested below.

First the proposed filtering method is described. The numbers in parentheses represent the steps of the algorithm shown in Figure 5.11. The initial step is to perform a clustering with a moderate amount of clusters (say 10 – 20 clusters) in either one or two dimensions (1). This process is repeated several times (2) in order to ensure that the largest few clusters found in (1) are “stable” in models with similar numbers of clusters. If the large clusters show up in the

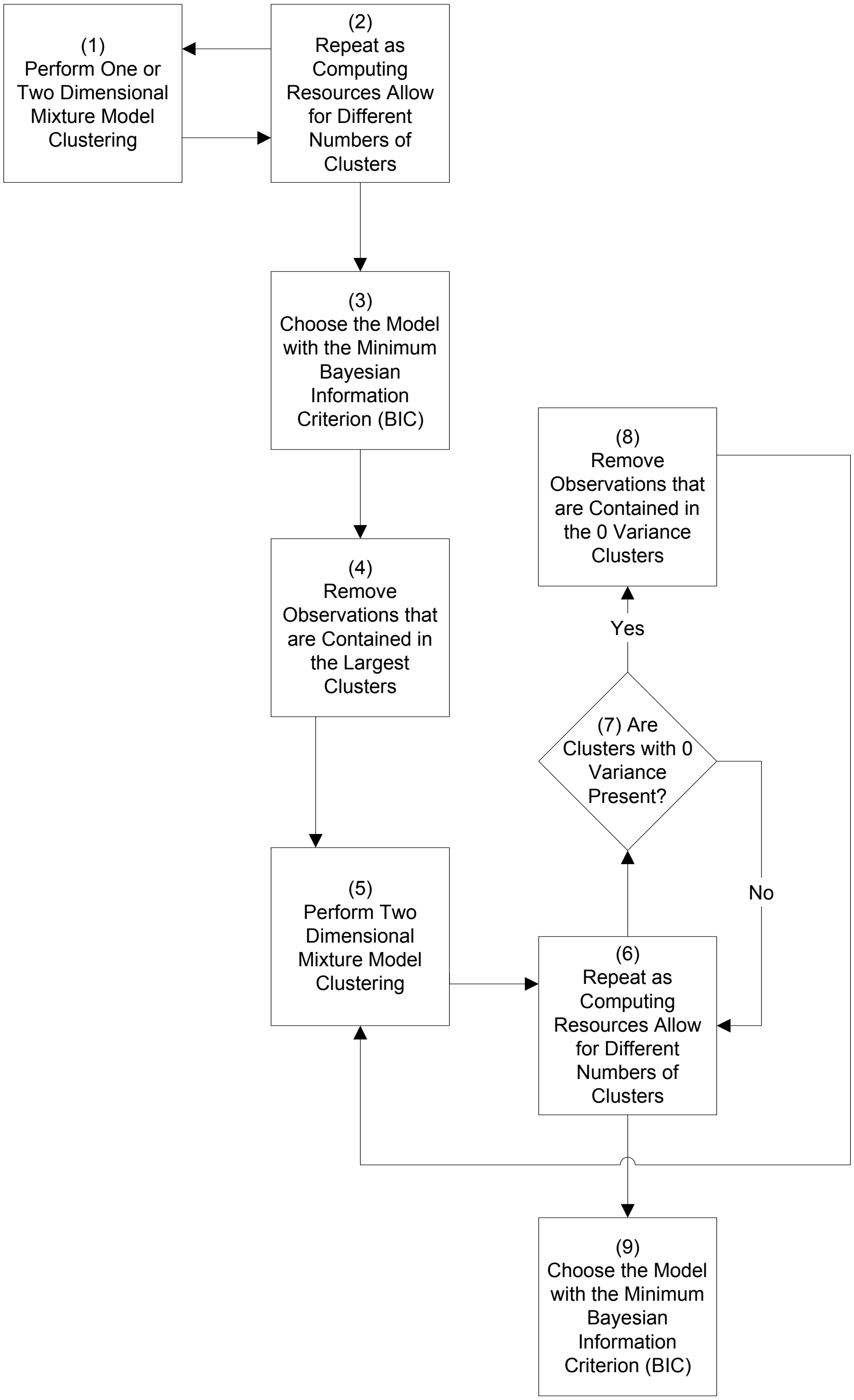

(1)
Perform One or Two Dimensional Mixture Model Clustering
(2)
Repeat as Computing Resources Allow for Different Numbers of Clusters
(3)
Choose the Model with the Minimum Bayesian Information Criterion (BIC)
(4)
Remove Observations that are Contained in the Largest Clusters
(5)
Perform Two Dimensional Mixture Model Clustering
(6)
Repeat as Computing Resources Allow for Different Numbers of Clusters
(7) Are Clusters with 0 Variance Present?
Yes
No
(8)
Remove Observations that are Contained in the 0 Variance Clusters
(9)
Choose the Model with the Minimum Bayesian Information Criterion (BIC)

Figure 5.11: Flowchart for Proposed Filtering for Two Dimensional Clustering

models in (1) and (2), the observations contained in the largest few clusters for the model having the minimum BIC value (3) are removed from the data set (4). The logic behind this is that large clusters most likely contain observations which are noisy and may not be informative. These large clusters typically have small means and may have small variances as well. The mixture model algorithm attempts to split these noisy clusters which considerably slows down the overall convergence rate and produces more clusters with little added information. The model usually runs more quickly with these observations removed and may be more likely to find moderately sized clusters of observations which may be of more interest to the user.

Once the data are filtered using either the one or the two dimensional clustering algorithm (the two dimensional technique is preferable but not always computationally feasible due to the large numbers of observations that are typically present), the two dimensional clustering algorithm is run (5). This is repeated for models having different numbers of clusters (6). The clusters are examined to see if there are any clusters that have zero variance (7). This situation occurs rarely and means that the observations contained in the cluster have identical values. If this is the case, the observations contained in these clusters are removed since they cannot be put into any "better" clusters (8). After removing the observations, the two dimensional clustering algorithm is restarted (5). The algorithm must be restarted anytime that observations are removed because models with different numbers of observations do not have comparable likelihoods or information criteria (AIC, BIC) . Once "enough" models have been run, the final model is selected based on the minimum BIC value (9). For microarray data, it is often difficult to find models having the number of clusters that "minimize" the BIC.

This is due to the large amount of computational resources necessary to run models having large numbers of observations and/or clusters. Once the BIC value approaches an asymptote, the model containing the number of clusters at which the slope flattens (or at the point of inflection) could be chosen. Models having higher numbers of clusters may have a slightly better fit but may not be worth the tradeoff between computer time and marginally improved cluster results.

The data from Ross *et al.* (2000) is filtered and analyzed using the method suggested in Figure 5.11. First, a two dimensional cluster analysis was performed on these data using k-means starting values. Models with 10, 15, and 20 clusters were fitted. In each of these models, there was a large cluster present which contained between 30 and 40 percent of the observations. Since the largest cluster appears to be stable in the models (10, 15, and 20 clusters) examined, observations falling in this cluster (bolded in Table 5.15) are removed and the filtered data is reanalyzed. The results of the 10 cluster model shown in Table 5.15 are used for filtering.

Table 5.15: Parameter Estimates for the 10 Cluster Model

Proportion	Mean	Variance
0.353	**0.0001**	**0.0545**
0.214	-0.0088	0.0110
0.185	0.0919	0.0179
0.160	-0.0011	0.0207
0.057	-0.2225	0.3240
0.012	0.1388	0.0003
0.008	0.2021	0.0006
0.008	-0.2424	0.0014
0.001	-0.7047	0.0022
0.001	0.7145	0.0044

Notice in Table 5.15 that the largest cluster (bolded) has a small mean. Observations with small expression values are often non-informative. Thus, this cluster may contain many observations that are attributable to noise. Therefore, removing the observations that are contained in this cluster may help to improve the signal to noise ratio and lead to better estimates and faster convergence.

The large cluster contains 9,099 observations (number genes x number samples) which are removed. The new data set has 16,641 observations. Estimation proceeds starting with the 10 cluster model (Table 5.16).

Table 5.16: Fit Statistics for the Two Dimensional Normal Mixture Model Applied to the Ross Data: 16,641 Observations

Number of Clusters	Log Likelihood	AIC	BIC
10	9099.49	-18140.98	-17917.11
15	8862.63	-17637.27	-17297.60
20	8912.93	-17707.87	-17252.41

One of the clusters in the 25 cluster model has an actual variance of zero. This means that the gene expression values were identical for this cluster. Such clusters are not informative and are removed from the model as they occur from this point forward. There were 173 observations removed (listed in Appendix 5.6) because they were members of the zero variance cluster. This means that the likelihood, AIC, and BIC values are no longer comparable since the number of observations has changed. The expression values for the observations falling in this cluster are 0.13033377. Observing this many identical observations recorded with 8 decimal digit accuracy is unlikely and may indicate that the reported expression values were actually observed at a lower precision. This could have also occurred due to an incorrect number of significant digits being retained after transforming the data. The estimation begins anew with the 10 cluster model with 16,468 observations with the filtered observations removed (Table 5.17).

Table 5.17: Fit Statistics for the Two Dimensional Normal Mixture Model Applied to the Ross Data: 16,468 Observations

Number of Clusters	Log Likelihood	AIC	BIC
10	6982.54	-13907.08	-13683.52
15	7214.30	-14340.60	-14001.40
20	7905.27	-15692.53	-15237.70
25	7414.84	-14699.68	-14198.59
⋮	⋮	⋮	⋮
100	7902.75	-15183.50	-12785.98

The log likelihoods oscillate somewhat for different numbers of clusters (Table 5.17). The models are not nested because the clusters in a model having a certain number of clusters cannot necessarily be reproduced by combining clusters from a model containing a larger number of clusters. For example, consider the density for the two cluster case shown in Equation 5.45 and the density for the three cluster case shown in Equation 5.46.

$$f_{y_i} = \pi\phi\left(\mu_1,\ \sigma_1^2\right) + (1-\pi)\phi\left(\mu_2,\ \sigma_2^2\right) \tag{5.45}$$

$$f_{y_i} = \pi_1\phi\left(\mu_1,\ \sigma_1^2\right) + \pi_2\phi\left(\mu_2,\ \sigma_2^2\right) + (1-\pi_1-\pi_2)\phi\left(\mu_3,\ \sigma_3^2\right) \tag{5.46}$$

If $\pi_2 = 0$ in Equation 5.46, the three component mixture becomes a two component mixture like the one shown in Equation 5.45. However, any observations that were actually from the $\phi\left(\mu_2,\ \sigma_2^2\right)$ distribution must be assigned to one of the other two clusters. If all of these observations were assigned to only one of the other clusters, the model would be nested.

However, there is no guarantee that this will occur and thus there is no guarantee that the models are nested.

Since the models are not nested, the likelihood may not be strictly monotonic. The general trend of the likelihood function for increasing values of *C* follows an expected pattern. That is, it increases. However, as previously stated, the likelihood function may oscillate for consecutive values of *C*. This is mainly due to convergence issues. In practice, an analyst may plot the likelihoods and look for the point at which the likelihoods begin to taper. The model containing the number of clusters at which such an asymptote occurs may be selected as the model with the "optimal" number of clusters.

Appropriate starting values become more difficult to select as the number of clusters increases. The convergence time also increases significantly with the number of clusters. It is our experience (Chapter 4) that microarray data tend to have large numbers of clusters present. Thus, a model containing 100 clusters was run. This model took several hours to run on a high speed Unix workstation with few competing jobs. The BIC is much lower for this model than for the 10 – 25 cluster models. Ideally, many different models would be run and the BIC would be plotted to determine where it reaches an asymptote. However, such an approach is not feasible due to limited computing resources. The "true" number of clusters is likely significantly larger than 100. This model could provide useful clusters, even though it may not be the "optimal" model.

The results of the 100 cluster model were first ranked from lowest to highest variance and then ranked from highest to lowest mean. The ranking is done in this manner because clusters with higher means and lower variances are more likely to be informative since they are being expressed at higher levels. The parameter estimates for the 50 top ranked clusters (according to this method) from the 100 cluster model are given in Table 5.18.

Table 5.18: Parameter Estimates for the Top 50 Ranked Clusters from the 100 Cluster Model

Number	Proportion	Count	Mean	Variance
1	0.00018219	3	-0.01322830	0.00000001
2	0.00018219	3	-0.01772880	0.00000001
3	**0.00176120**	**29**	**0.11897956**	**0.00000282**
4	0.00133609	22	0.11907273	0.00000283
5	0.00103243	17	0.11901975	0.00000289
6	0.00030366	5	-0.01682870	0.00000405
7	0.00072877	12	-0.01472840	0.00000491
8	0.00024292	4	-0.01660360	0.00000506
9	0.00121462	20	-0.01570350	0.00000528
10	0.00042512	7	-0.01580000	0.00000579
11	0.00030366	5	-0.01502850	0.00000608
12	0.00018219	3	-0.01622860	0.00000662
13	0.00018219	3	-0.01472840	0.00000675
14	0.00024292	4	-0.01547850	0.00000675
15	0.00018219	3	-0.01622860	0.00000675
16	0.00297583	49	0.11948945	0.00000781
17	0.00157901	26	0.11993022	0.00000949
18	0.00091097	15	0.04295527	0.00001285
19	0.00248998	41	0.11999555	0.00001446
20	0.00419045	69	0.04262931	0.00001795
21	0.00425118	70	0.12170162	0.00004005
22	0.00115389	19	0.00470677	0.00006149
23	0.00340095	56	0.00377446	0.00007608
24	0.00248998	41	0.00517476	0.00007969
25	0.00394753	65	-0.04162680	0.00013590
26	0.00261144	43	-0.03978880	0.00014267
27	0.00097170	16	-0.23425220	0.00044511
28	0.00121462	20	-0.23083700	0.00053594
29	0.00097170	16	-0.23687100	0.00071334
30	0.00048585	8	-0.22449200	0.00083003
31	0.00048585	8	-0.22749190	0.00109679
32	0.00072877	12	-0.23844250	0.00118085
33	0.00072877	12	0.08592268	0.00368708
34	0.00097170	16	0.17897769	0.07394535
35	0.00091097	15	-0.16045030	0.10694560
36	0.00060731	10	-0.11224170	0.14280320

Table 5.18: Parameter Estimates for the Top 50 Ranked Clusters from the 100 Cluster Model (continued)

Number	Proportion	Count	Mean	Variance
37	0.00267217	44	-0.08861490	0.15812844
38	0.00078951	13	-0.11782530	0.16888789
39	0.00115389	19	-0.08862410	0.18876703
40	0.00042512	7	0.15229334	0.19527841
41	0.00078951	13	-0.12256280	0.20491500
42	0.00085024	14	-0.06092030	0.22709623
43	0.00085024	14	-0.05321190	0.24540903
44	0.00394753	65	-0.10954510	0.25338310
45	0.00036439	6	-0.01701320	0.27409305
46	0.00030366	5	0.22560336	0.29605262
47	0.00078951	13	0.05418408	0.29834202
48	0.00030366	5	0.00691714	0.31286753
49	0.00018219	3	0.38069299	0.47621969
50	0.00054658	9	0.25091027	0.53135585

If the assumptions for the normal mixture model hold, then a normal density should fit well for the individual clusters. Many of the variances in Table 5.18 are quite small. Small variances result in a peaked normal density with a small spread. As the number of clusters increases, clusters such as these are more likely to occur. Clusters one and two have only three observations and thus are likely not useful for a diagnostic tool. Table 5.19 shows the cluster members for cluster number three reported in Table 5.18.

Observe in Table 5.19 that there are several groups of observations that have identical expression values. Such a result is unlikely (as discussed above). However, such observations do tend to fall in the same distribution and thus the normal mixture model based clustering places them into the same cluster. If the identical observations had fallen into the same cluster, they would have been removed in the filtering process. The density for cluster three is shown in Figure 5.12.

Table 5.19: Cluster Members for Cluster Number 3

Gene	Cell Line	Expression Value
W79419	1	0.12057393
AA057359	2	0.12057393
AA046312	3	0.1172713
AA056232	5	0.12057393
AA045090	6	0.12057393
H26976	8	0.12057393
N93479	10	0.12057393
AA025275	12	0.1172713
N72074	13	0.1172713
AA026222	15	0.1172713
W86907	18	0.1172713
AA046316	19	0.1172713
AA035619	21	0.12057393
W46479	23	0.1172713
AA040777	23	0.12057393
AA047408	25	0.1172713
AA031493	31	0.12057393
AA056232	31	0.12057393
AA047408	33	0.12057393
R16808	33	0.1172713
H85095	35	0.1172713
AA057728	35	0.12057393
R59373	37	0.1172713
AA057728	38	0.12057393
AA047268	43	0.1172713
H95849	47	0.12057393
AA039599	52	0.1172713
N48061	55	0.12057393
AA042879	60	0.1172713

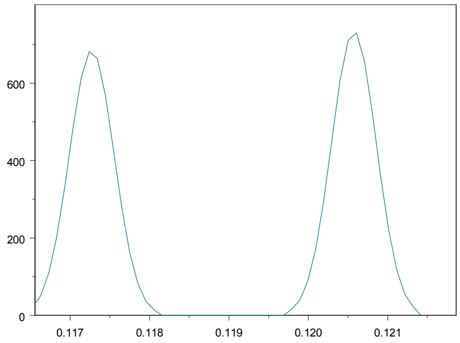


Figure 5.12: Density Plot for Cluster Three

There are two distinct modes present in Figure 5.12. This is due to the two groups of similar valued observations shown in Table 5.19. This cluster would readily divide into two clusters if a model with a larger number of clusters were fitted.

The density for the 16,468 observation filtered data set (Table 5.17) is shown in Figure 5.13.

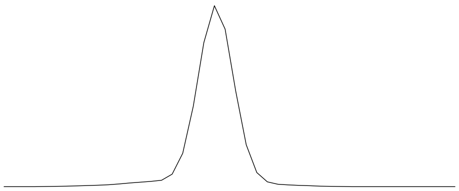


Figure 5.13: Density Plot for 16,468 Observations

Figure 5.13 looks similar to a normal distribution except for the wide tails. The observations contained in these tails may be noisy and one may wish to consider a filtering method which removes them. However, there may be modes that are not visible in the tails which could represent clusters.

The plot was generated by fitting a normal density to a histogram having 100 "bars". There are modes present that are not visible at the current resolution. Increasing the number of "bars" for the histogram results in plots with wider tails. Since the mixture density (Figure 5.13) closely resembles the normal density, the estimation becomes more difficult than it would be with a more multimodal mixture density. The mean and variance for the data are -0.0009652 and 0.04650166, respectively.

For the normal mixture model, the observations in an individual cluster are expected to come from the same normal distribution. The densities for several of the clusters are shown below. Figure 5.14 shows the density for a cluster of size 806 which is similar to a normal density, although the tail on the right contains a “bump” which could represent observations that would be included in a separate cluster in a model having more than the 100 clusters fitted here. The mean for this cluster is 0.08597038 and the variance is 0.00096818. This illustrates that the two dimensional clustering algorithm puts together clusters based on the expression values belonging to the same distribution. Such information may be helpful in a diagnostic setting because genes and cell lines from a distribution of expression values may be indicative of the biological “state” of the cell, that is observed with random error. The expression values for the genes and cell lines in the medium sized clusters could be used to develop assays for typing an unknown cell. This could be particularly useful for identifying similar types of tumor cells.

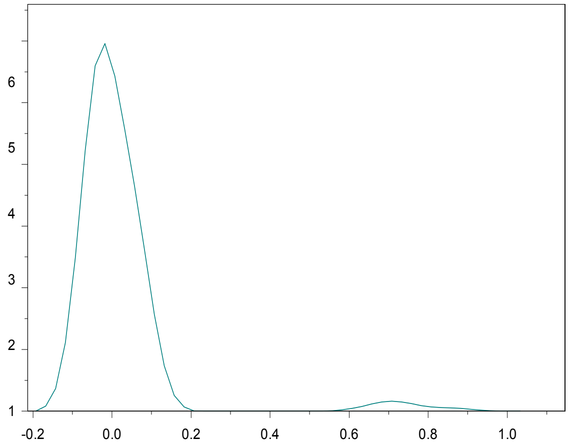


Figure 5.14: Density Plot for Two Dimensional Cluster of Size 806

Other clusters such as the one shown in Figure 5.15 are also similar to the normal density (unimodal) except for the “chatter” at the top of the density. Such chatter may be indicative of other sub-clusters. This cluster has 941 observations, and a mean and variance of 0.05936204 and 0.01851001, respectively. The variance is higher for this cluster, and provides a further indication that other sub-clusters may exist.

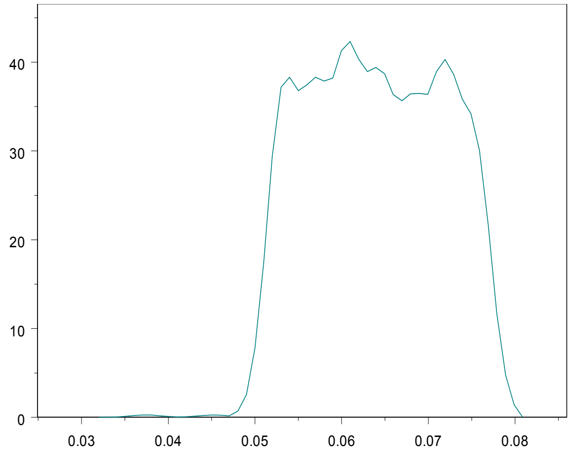


Figure 5.15: Density Plot for Two Dimensional Cluster of Size 941

Some clusters such as the one shown in Figure 5.19 exhibit modes that are well separated. The mean and variance for this cluster are 0.07769306 and 0.00413041, respectively. The variance is still quite small for a cluster of size 1365. Two well separated modes are visible in Figure 5.16, and this cluster could be readily split into two separate clusters.

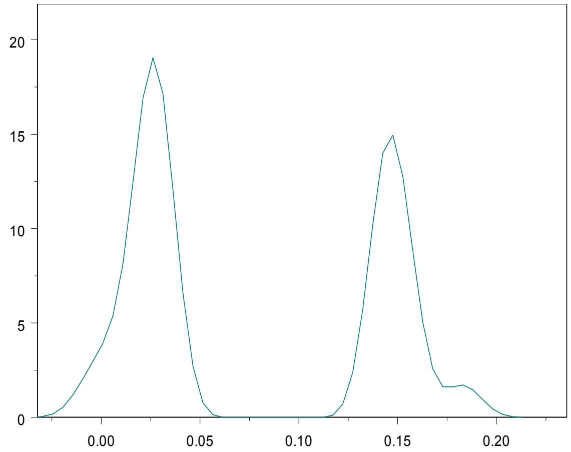


Figure 5.16: Density Plot for Two Dimensional Cluster of Size 1365

Finally, there are large clusters that clearly need to be split into multiple clusters. Such a cluster is shown in Figure 5.17.

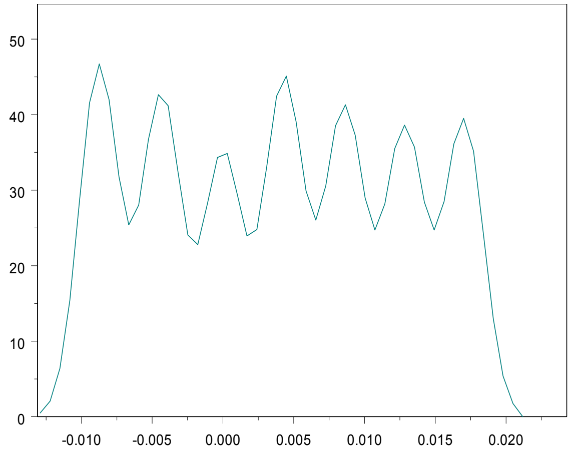


Figure 5.17: Density Plot for Two Dimensional Cluster of Size 1175

The cluster in Figure 5.17 has 1175 observations and a mean and variance of 0.00387388 and 0.00007562, respectively. The mean and variance are both quite small for this cluster, indicating that the members of this cluster have small expression values. Such a result may arise from groups of "noisy" genes. Numerical precision could also be a factor in the inability to separate these observations in the 100 cluster model. It is apparent from Figure 5.20 that this cluster could be further subdivided. However the need for this may be questionable since the range of the observations seems negligible.

The two dimensional clustering algorithm groups observations that are likely to have come from the same normal distribution. For microarray data, this often yields clusters which are "tight" in the sense that they have small variances. It is difficult to discover gene pathways using this approach because observations with numerically similar gene expression values do not necessarily fall on the same pathways. Therefore, we suggest that the two dimensional clustering technique be applied to microarray data for which the goal is to develop assays or diagnostic tests. Groups of genes and cell lines which cluster together may be useful in typing an unknown cell based on its gene expression levels.

5.2.7 Interpreting the Cluster Results

Interpreting the cluster results for the two dimensional normal mixture model clustering algorithm is unlike interpreting the cluster results of other techniques. Each observation has two indices associated with it. For the Ross *et al.* (2000) microarray data, one of these indices is the gene identifier and the other index is the cell line number. The cluster members for a cluster from the Ross data set analysis are shown in Table 5.20. The mean of the observations in this cluster is -0.0886241 and the variance is 0.18876703.

Table 5.20: Cluster Members for a 19 Observation Cluster

Gene	Cell Line	Tumor Type	Expression Value
AA054290	1	Non-Small Cell Lung Cancer	1.11058971
AA026207	2	Non-Small Cell Lung Cancer	-0.229148
AA031671	2	Non-Small Cell Lung Cancer	-0.229148
AA036724	5	Non-Small Cell Lung Cancer	-0.236572
W92696	7	Non-Small Cell Lung Cancer	-0.2596373
AA040878	8	Non-Small Cell Lung Cancer	-0.19382
AA045192	9	Non-Small Cell Lung Cancer	-0.251812
AA026796	10	Colon Cancer	-0.2076083
AA044233	11	Colon Cancer	-0.2146702
R83277	15	Colon Cancer	-0.2676062
W93802	19	Breast Cancer	-0.19382
H50438	23	Breast Cancer	-0.2218488
AA046035	24	Breast Cancer	1.17318627
R83277	32	Leukemia	-0.229148
R11631	34	Melanoma	-0.2676062
W42423	43	Leukemia	-0.2441251
W78128	46	Renal Cancer	-0.2596373
AA054226	56	Central Nervous System Cancer	-0.19382
H65189	58	Central Nervous System Cancer	-0.2676062

Notice that the observations colored in pink in Table 5.20 are from the same gene (R83277) but different cell lines (15 and 32). The observations colored in green are from the same cell line (2) but different genes (AA026207 and AA031671) and have the same gene expression value (-0.229148). There are 18/429 genes and 18/60 cell lines represented in this cluster. Duplicates for either the cell lines or the genes in a single cluster are expected due to the two dimensional nature of the clustering and may represent a biologically interesting reason for this combination of genes and cell lines clustering together.

Figure 5.21 shows the density plot for the 19 observation cluster. There are two modes apparent in this figure. The mode on the left is more prominent in the sense that it contains more observations. Such a cluster could be split in a model with more clusters. The bimodal nature of Figure 5.18 may be due in part to the small cluster size of 19.

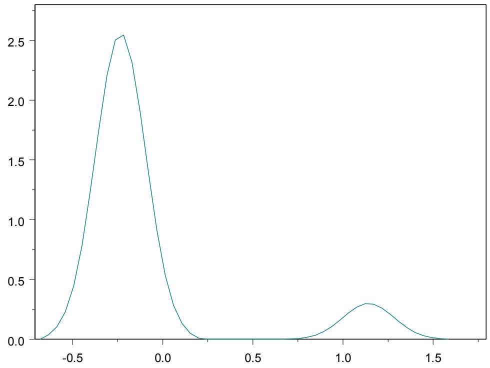


Figure 5.18: Density Plot for a 19 Observation Cluster

5.2.8 Analysis of a Selected Subset of the Ross *et al.* (2000) Data Set

In order to further evaluate our method using microarray data, a selected subset (chosen by Dr. Windle) of the Ross *et al.* (2000) data was analyzed. This data set contains 77 genes for 60 cell lines, or a total of 4,620 observations. 21 of these genes are known to be in the melanin pathway. There are a total of 8 melanoma cells present in the 60 cell lines. The melanin genes are typically highly expressed when present. I was blinded as to the identities of the cell and gene labels. A density plot for this data subset is given in Figure 5.19.

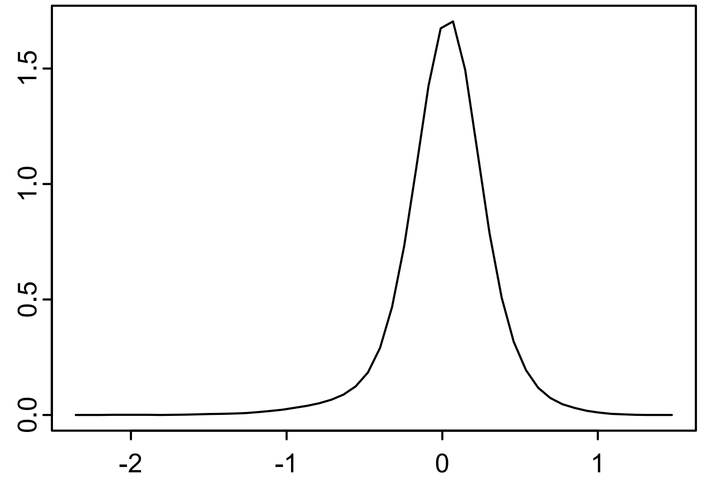


Figure 5.19: Density Plot for 77 Gene Selected Subset

The mean of the observations is 0.021 and the variance is 0.071. There may be modes that are not visible at this scale, particularly in the long left tail.

An eleven cluster model was chosen based on the cluster density plots being unimodal in appearance. (The BIC model selection criteria suggested a three cluster model, but these clusters were too large to be readily interpretable. These large clusters may serve as a starting point for generating further sub-clusters.) The parameter estimates for the eleven cluster model are given in Table 5.21. The clusters are sorted by mean. Notice that cluster 7 contains 52 percent of the observations and has a small mean (-0.082). Once again, observations in such a cluster would be removed if the filtering method presented in Figure 5.11 were applied. Clusters 2 and 5 are bolded because they were found to contain many of the genes in the melanin pathway and many of the melanoma cell lines. Notice that these clusters also have the largest mean expression values, which is consistent with the literature which suggests that melanin is highly expressed when present (Ross *et al.*, 2000).

Table 5.21: Parameter Estimates for 11 Cluster Model

Cluster #	N	Proportion	Mean	Variance
5	**87**	**0.019**	**0.723**	**0.01514**
2	**288**	**0.062**	**0.431**	**0.00392**
3	405	0.088	0.277	0.00122
9	549	0.119	0.172	0.00052
11	405	0.088	0.098	0.00015
10	273	0.059	0.010	0.00005
7	2418	0.523	-0.082	0.01835
6	23	0.005	-0.450	0.00004
8	26	0.006	-0.570	0.00014
1	127	0.027	-0.723	0.02476
4	19	0.004	-1.311	0.04847

The densities for clusters 2 and 5 are plotted in Figure 5.20.

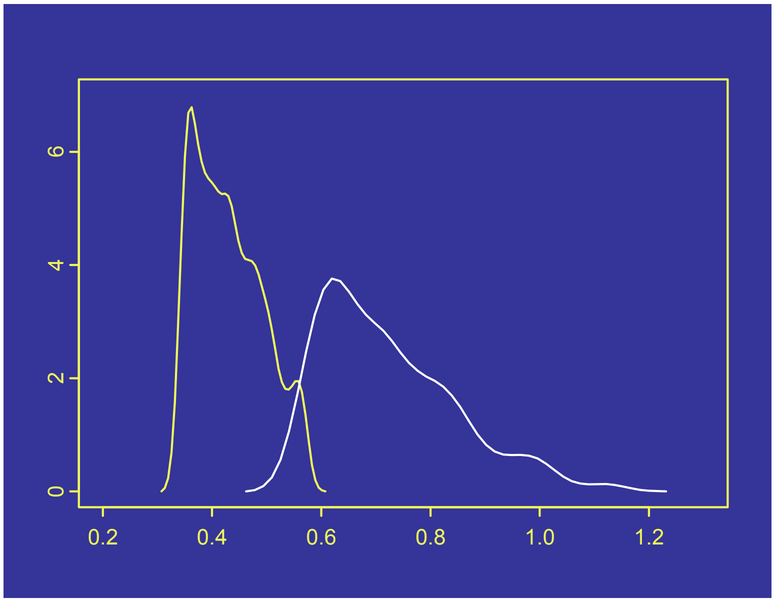


Figure 5.20: Density Plot for Clusters 2 and 5

Cluster 2 is shown in yellow in Figure 6.20, while cluster 5 is shown in white. Cluster 2 contains 288 observations and has a mean and variance of 0.431 and 0.004, respectively. Cluster 5 contains 87 observations and has a mean and variance of 0.723 and 0.015, respectively. Since these clusters are adjacent to each other and appear to be quite similar (Figure 6.20), it is possible that these two clusters need not have been split and can be combined to form a single cluster.

The second cluster contains 7 out of 8 of the melanoma cell lines and 16 out of 21 of the genes known to be active in the melanin pathway. The fifth clusters contains 2 out of 8 of the melanoma cell lines and 9 out of 21 of the genes known to be active in the melanin pathway. Thus, our method has correctly identified many (gene, cell line) pairs for the melanoma cells and the genes in the melanin pathway.

5.3 Conclusion

The advantage of using parametric models, such as the normal mixture model, for clustering is that one can formally evaluate the model fit by using likelihood based statistics. Normal mixture models for clustering require the number of clusters to be specified. Section 4.3.5 introduced three criteria for evaluating the fit of a mixture model and suggested the Bayesian Information Criterion (BIC) as the measure of choice. Alternative methods for evaluating the model fit are described in McLachlan and Peel (2000). Currently, multiple models having different numbers of clusters must be fit and the best fitting model selected using statistics such as the BIC. In order to lessen this workload, more research is needed on determining *a priori* how many clusters are expected. This problem is listed as a future research problem in Chapter 6.

In cases where large numbers of clusters are present, there are frequently a handful of very large clusters present. This circumstance could negatively affect the accuracy of the estimation for the numerous remaining smaller clusters, since the large clusters often contain a high degree of noise. One option recommended above for analyzing such data is to remove the observations which are members of large clusters and to re-cluster the remaining observations. This technique may help in focusing on signals of interest which are often contained in the smaller clusters. Another option for analyzing large data sets is to initially fit a mixture model containing a moderate number of clusters and then to re-cluster these clusters individually. Such an approach is faster than simply fitting a model having a large number of clusters initially, and the results will be identical.

The analyses of the two dimensional microarray data presented in this chapter do not require the data to be collapsed. By preserving the row and column indices, the clustering results maintain both dimensions of the data. The results are simultaneous clusters of rows and columns. For microarray data, such an analysis allows groups of genes and samples to be established. The results are often presented in the form of a color map, where the rows and columns indicate the row and the column from which the observation came and the color indicates the cluster. Color maps become difficult to interpret for large numbers of clusters, as it can be hard to distinguish similar shades of colors, and thus are not shown for the 100 cluster model of Section 5.2.6. An alternative is to examine lists of cluster members manually, which becomes tedious. There is a need, as suggested in Chapter 6, for more research on techniques for displaying and interpreting cluster results.

The two dimensional mixture model clustering algorithm has several advantages. It provides a measure of how well a model with a given number of clusters fits the data. The method is not sensitive to the ordering of the data, as many hierarchical clustering methods can be. Missing observations are easily handled, as each cluster has a normal distribution and for moderate sample sizes a missing observation will not strongly affect the parameter estimation. The summary statistics for one dimensional clustering methods that require summarizing the data in one dimension must be modified to handle missing observations. Missing observations may bias the results of such methods. The cluster assignments are flexible in the sense that each observation has a posterior probability of belonging to every cluster. Cluster assignments are typically made according to the maximum posterior probability. However, if the posterior probabilities of an observation belonging to several clusters are similar, cluster assignments may be based on a lower ranked posterior probability. Confidence intervals for these posterior probabilities may also be calculated and help to support the use of overlapping clusters. Posterior probability confidence intervals that overlap for a particular observation indicate that the observation may be more accurately assigned by allowing it to be a member of multiple clusters. Finally, the two dimensional mixture clustering does not require summarizing the data in either dimension, as discussed above.

The model based parametric clustering method proposed here differs from the commonly used clustering methods in that the underlying distribution is not uniform. Therefore, differences in magnitude in the observations contained in a single cluster are allowed. For a given cluster, the observations are assumed to come from the same normal distribution. This is different from methods that rank the genes based on their expression levels or selected effect sizes. While both assume normality, the assumptions for the mixture model are on the individual clusters rather than the sample distribution. The non-mixture model methods also treat differences in magnitude to be absolute and do not allow for random error.

There are a few issues that must be considered when applying the two dimensional mixture model clustering algorithm. The observations are assumed to come from a mixture of normal distributions. Each cluster is expected to be normally distributed. This may not be a reasonable assumption for some data. The algorithm requires starting values to be specified. Although the algorithm is robust with respect to starting values, choosing very poor values may result in the failure of the model to converge due to numerical precision issues, as well as potentially slow the convergence speed. Gene independence is another major assumption made and is discussed further in Chapter 6.

Chapter 6

Conclusions

6.1 Dissertation Summary

In this dissertation, a range of tools for the analysis of microarray data was presented. The first chapter introduced microarray technology, explained the computational challenges of analyzing microarray data, and discussed the historical development and application of clustering techniques.

In the second chapter, clustering techniques including similarity and distance measures were reviewed. A time course microarray experiment by Chu *et al.* (1998) was introduced. The need for filtering microarray data was explained. In order to improve the interpretability of the cluster results, data smoothing was performed using LOESS. Filtering was performed using a method based on Pearson's correlation between gene profiles and seven average profiles constructed from genes with known function. Clustering was performed using the average linkage and k-means clustering methods for both smoothed and unsmoothed data using two different filtering thresholds.

In chapter three, three methods for re-labeling clusters to assess the agreement between the results from different clustering techniques were proposed. Cluster agreement measures were discussed, including the kappa statistic and the Rand index. These measures were applied to evaluate how well average linkage clustering did versus a k-means clustering at reproducing known results. Three microarray experiments targeting the same cell lines and measuring many of the same genes were examined. Two of these experiments used Affymetrix designs (Millenium Pharmaceutical Company; Staunton *et al.*, 2001). The other experiment used the cDNA design (Ross *et al.*, 2000). Cluster analyses were performed on each of these data sets and the results were compared and discussed.

In chapter four, a parametric clustering technique based on mixture models was presented. The necessary theory was developed. The Expectation/Maximization (EM) algorithm was proposed for estimation. The hybrid EM/Newton-Raphson algorithm was suggested for improving the convergence in some situations. The Akaike information criterion (AIC) and Bayesian Information Criterion (BIC) were proposed as statistics useful for evaluating the model fit. Confidence interval formulas were derived for the parameters. Simulations were run in order to evaluate the ability of the EM algorithm to recover the parameter estimates. Finally, the microarray data set from Ross *et al.* (2000) was analyzed and discussed.

In chapter five, the methods developed in chapter four were extended to support the analysis of two dimensional data. The motivation behind two dimensional clustering applications was discussed. The theory was extended to the two dimensional case. A detailed discussion was given on the possible restrictions on the proportions (marginal versus global). Data was simulated and analyzed. Finally, the microarray data set from Ross *et al.* (2000) was analyzed and discussed.

A user friendly program, 2-DCluster, was written to support these methods. This program was written for Microsoft Windows 2000 and XP systems and supports one and two dimensional univariate clustering. The program and sample applications are available at http://etd.vcu.edu. An electronic copy of this dissertation is available at the same address.

6.2 Other Approaches to Clustering

There are several other approaches frequently used for the clustering and classification of microarray data. Data mining approaches for clustering and classification are among the most popular. Such approaches generally make no assumptions regarding the distribution of the data, and therefore are not model based. These include techniques such as self organizing maps and genetic algorithms which are frequently used in data mining applications.

Self organizing maps (SOM) are an extension of "neural networks" (Gurney, 1997; Bohr, 2003). To implement an SOM, a geometry of nodes must first be chosen (say a 2 x 2 grid). The nodes are initially mapped into the data space at random. The location of the nodes is iteratively adjusted by randomly selecting a data point and then moving the nodes in the direction of that data point. Once the algorithm has proceeded through a user defined number of iterations, the algorithm terminates with "similar" data points grouped around a specific node. Extracting clusters from SOMs requires selecting a boundary around each node (or set of nodes) in order to define a cluster. For more information on self organizing maps applied to microarray data analysis, see Tamayo *et al.* (1999), Toronen *et al.* (1999), Herrero *et al.* (2001), and Wang *et al.* (2002).

Genetic algorithms are modeled on the biological processes of natural selection and evolution. A function to be optimized, called a fitness function, must be defined.

The genetic algorithm (GA) maximizes this function by adjusting the values of the variables involved in the function. A population of possible solutions is maintained, with each member of the population containing a set of values for the variables. The fitness function is evaluated for every population member. The members achieving the highest fitness function values are kept and used as “parents” for generating the other members of the population. The “children” are generated by choosing values for the variables based on some combination of the values from the successful parents. (This process is similar to the actions of natural selection and evolution.) Cluster analysis using genetic algorithms could be done by specifying, for example, a mixture likelihood function to be optimized. Many other approaches for using GAs for clustering or classification are possible. For more information on genetic algorithms, see Holland (1975), Goldberg (1989), or Harvey (2001).

6.3 Extensions and Suggestions for Future Research

This section contains suggestions for applications which extend the results of this dissertation and suggests possible problems for future research. Such applications may be useful for microarray data analysis, as well as being helpful in other fields. Several areas of cluster analysis need further study. Some of these areas are estimating the number of clusters prior to the analysis, normalizing the data, presenting clustering results graphically for multidimensional data, and filtering the data in order to reduce the noise and the computational requirements. More specific suggestions are given in this section.

6.3.1 The Relative Contribution of Genes and Environment in Behavioral and Mental Disorders

Understanding the genetics of behavioral and mental disorders is a difficult problem. In addition to many genes being involved in these traits, there are often complex environmental factors present. Extracting the relative contributions of genetic and environmental influences for a given trait requires genetically informative data. One of the best approaches to this problem makes use of twin data. The two types of twins are monozygotic and dizygotic. Monozygotic twins are expected to be genetically identical, while dizygotic twins are expected to share half of their genes on average. This knowledge may be used to build models which estimate the relative contribution of genes and environment for a given phenotype. However, there is no easy way to verify these results biologically or to determine which genes are involved.

Neale (2003) fitted a two component bivariate mixture to twin data. One component is for the monozygotic (MZ) twins, while the other is for the dizygotic (DZ) twins. Each component has a two dimensional covariance matrix associated with it, with one observation coming from each twin. The covariance matrix for MZ twins is given in Equation 6.1.

$$\Sigma_{MZ} = \begin{bmatrix} a^2 + c^2 + e^2 & a^2 + c^2 \\ a^2 + c^2 & a^2 + c^2 + e^2 \end{bmatrix}, \qquad (6.1)$$

where *a*, *c*, and *e* are the components of variance due to additive genetic, common environmental, and specific environmental factors, respectively. Similarly, the covariance matrix for DZ twins is given in Equation 6.2.

$$\Sigma_{DZ} = \begin{bmatrix} a^2 + c^2 + e^2 & 0.5a^2 + c^2 \\ 0.5a^2 + c^2 & a^2 + c^2 + e^2 \end{bmatrix} \tag{6.2}$$

Neale (2003) recovers the usual additive genetic, common environment, and specific environment estimates by fitting a mixture model using likelihood based techniques. The mixture density is:

$$\begin{aligned} F_i &= p(MZ_i)f(MZ_i) + p(DZ_i)f(DZ_i) \\ &= p(MZ_i)f(MZ_i) + \left[1 - p(MZ_i)\right]f(DZ_i), \end{aligned} \tag{6.3}$$

where *i* represents a specific twin pair and $p(MZ_i)$ and $p(DZ_i)$ represent the probability of the i^{th} twin pair being monozygotic and dizygotic, respectively. Twin zygosity is not perfectly diagnosed, as such diagnoses are generally based on surveys rather than a blood test. If the true zygosity were known, then $p(MZ) = 1$ and $p(DZ) = 1$. Neale (2003) runs several models with different zygosity misclassification rates specified. In other words, only the covariance terms given in Equations 6.1 and 6.2 are estimated for fixed misclassification rates. Neale demonstrates that the mixture model approach performs quite well for zygosity misclassification rates of 15 percent or less.

Neale's (2003) approach requires the specification of the estimated zygosity misclassification rates. Once these rates are specified, the variance components are estimated using structural equation models (Neale and Cardon, 1992). The zygosity misclassification rates could be estimated by extending the structural equation model using the Expectation/Maximization (EM) algorithm. Initial values would be specified for the zygosity misclassification rates and the variance components. A posterior probability (similar to that described in Chapter 4) of each observation belonging to both clusters could then be calculated in the Expectation step. The posterior probability estimates could then be used to obtain the maximum likelihood estimates of the variance components. The structural equation model could be run using these estimates to obtain the maximum likelihood estimates for the variance components. This process would iterate until convergence was obtained.

As microarrays become cheaper, libraries containing DNA samples from twins will be built. Two dimensional clustering approaches could be applied to cluster genes and twins. Models could be built to combine the twin zygosity data with the clustering results. Such an approach could help to determine not only the relative contributions of genes and environment, but could also indicate which genes are operating in concert.

6.3.2 Evaluating the Genetic Effects of Complex Mixtures of Pollutants

Organizations such as the Environmental Protection Agency (EPA) are interested in studying complex mixtures of pollutants in the soil, air, and water. Assuming an appropriate experimental design, data may be collected on multiple toxicity measures and their effects on gene expression over a period of time. For each toxicity measure, a polynomial could be fitted to each gene across the time points. The coefficients of the polynomial could then be clustered using a multivariate clustering technique.

The toxicity measures reflect an underlying mixture of pollutants. A two dimensional multivariate mixture model could be applied here. One dimension of the data would be the genes and the other would be the toxicity measures. The model is multivariate due to clustering the polynomial coefficients described above. Such an approach would result in two dimensional clusters of genes and toxicity measures and may give regulators more information than a traditional linear model based analysis.

6.3.3 Combining Chemosensitivity Data with Gene Expression Data

There is a growing body of microarray data on chemosensitivity (Ross *et al.*, 2000). These experiments are designed to collect gene expression data on samples (such as tumor cells) exposed to specific chemicals. The two dimensional clustering method developed in this dissertation could be extended to support three dimensional clusters of genes, samples, and chemical exposures. The likelihood for such a normal mixture model could be written as:

$$L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)=\prod_{i=1}^{N_1}\prod_{j=1}^{N_2}\prod_{k=1}^{N_3}\sum_{q=1}^{C_1}\sum_{r=1}^{C_2}\sum_{s=1}^{C_3}\pi_{qrs}\phi\left(y_{ijk};\ \tau_{\mathrm{qrs}},\sigma_{qrs}^2\right), \tag{6.4}$$

where $i=1,\ldots,N_1$, $j=1,\ldots,N_2$, and $k=1,\ldots,N_3$ are the indices for the three dimensions, $q=1,\ldots,C_1$, $r=1,\ldots,C_2$, and $s=1,\ldots,C_3$ are the indices for the clusters ($qrs=1,\ldots,C_1C_2C_3$ for the global constraint on the proportions), y_{ijk} is the observation, and π_{qrs}, τ_{qrs}, and σ_{qrs}^2 are the cluster specific proportions, means, and variances, respectively. The log likelihood is given in Equation 6.5.

$$\ell\left(\mathbf{y};\ \boldsymbol{\pi}, \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right) = \log\left[L\left(\mathbf{y};\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)\right]$$

$$= \sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\sum_{k=1}^{N_3} \log\left[\sum_{q=1}^{C_1}\sum_{r=1}^{C_2}\sum_{s=1}^{C_3} \pi_{qrs}\phi\left(y_{ijk};\ \tau_{\mathrm{qrs}}, \sigma_{qrs}^2\right)\right] \tag{6.5}$$

The development of the theory proceeds similarly to that described in Chapter 5. This approach could aid researchers in discovering chemicals which interrupt gene activity for genes that may be involved in biological processes (such as the regulation of cancer growth).

6.3.4 One Dimensional Normal Mixture Models with Repeated Measures

The one dimensional normal mixture model was introduced in Chapter 4. When data has repeated measures on the observations to be clustered, the one dimensional normal mixture model requires the data for each observation to be collapsed into a single value. The usual method for collapsing is to take the mean of the repeated measurements for each observation. However, such an approach does not take into account the variability of the measurements and thus loses information which could improve the clustering results.

One approach which takes this variability into account requires rewriting the mixture likelihood to include the repeated observations, which all come from the same distribution. The normal mixture model distribution for repeated measures is:

$$f_{y_{ij}}\left(y_{ij};\ \boldsymbol{\pi}, \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right) = \sum_{k=1}^{C} \pi_k\ \phi\left(y_{ij};\ \mu_k,\ \sigma_{\mathrm{k}}^2\right), \tag{6.6}$$

where the parameters are as defined in Chapter 4, i refers to the observation number, j refers to the replicate number, and k refers to the cluster number. The likelihood for N observations and R replicates is:

$$L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)=\prod_{i=1}^{N}\prod_{j=1}^{R}\sum_{k=1}^{C}\pi_k\ \phi\left(y_{ij};\ \mu_k,\ \sigma_{\mathrm{k}}^2\right). \tag{6.7}$$

The log likelihood for the N observations and R replicates with the restriction that $\pi_C = 1-\sum_{k=1}^{C-1}\pi_k$ is:

$$\begin{aligned}\ell\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right) &= \log\left[L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)\right]\\ &=\sum_{i=1}^{N}\sum_{j=1}^{R}\log\left[\sum_{k=1}^{C}\pi_k\ \phi\left(y_{ij};\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)\right]\\ &=\sum_{i=1}^{N}\sum_{j=1}^{R}\log\left[\sum_{k=1}^{C-1}\pi_k\ \phi\left(y_{ij};\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)+\right.\\ &\qquad\left.\left(1-\sum_{k=1}^{C-1}\pi_k\right)\phi\left(y_{ij};\ \mu_C,\ \sigma_{\mathrm{C}}^2\right)\right].\end{aligned} \tag{6.8}$$

The posterior probability estimate is:

$$\alpha_{ik}=\frac{\hat{\pi}_k\ \sum_{\mathrm{j}=1}^{\mathrm{R}}\phi\left(\ y_{ij};\ \hat{\mu}_k,\ \hat{\sigma}_{\mathrm{k}}^2\right)}{\sum_{r=1}^{C}\hat{\pi}_r\ \sum_{\mathrm{j}=1}^{\mathrm{R}}\phi\left(\ y_{ij};\ \hat{\mu}_r,\ \hat{\sigma}_{\mathrm{r}}^2\right)}, \tag{6.9}$$

where *k* refers to the cluster number. The proportion estimate is:

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} \hat{\alpha}_{ik}}{N}. \tag{6.10}$$

The mean estimate is:

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} \sum_{j=1}^{R} y_{ij} \hat{\alpha}_{ik}}{R \sum_{i=1}^{N} \hat{\alpha}_{ik}}. \tag{6.11}$$

The variance estimate is:

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \sum_{j=1}^{R} \hat{\alpha}_{ik} \left(y_{ij} - \hat{\mu}_k \right)^2}{R \sum_{i=1}^{N} \hat{\alpha}_{ik}}. \tag{6.12}$$

Finally, the estimate for homogeneous variances is:

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{C} \hat{\sigma}_k^2 \left(\sum_{i=1}^{N} \hat{\alpha}_{ik} \right)}{N}. \tag{6.13}$$

Studies comparing the results of clustering using the repeated measures normal mixture model and the normal mixture model using different methods of collapsing the data need to be performed. Implementation issues such as convergence time should be examined for the normal mixture model using collapsed data versus the repeated measures normal mixture model. Such studies may give the user an idea of the tradeoffs in convergence time and the accuracy of results between clustering repeated measures data using summary statistics versus performing clustering using a method which directly supports repeated measures.

6.3.5 Data Filtering Approaches

As previously discussed, gene filtering methods play a vital role in the analysis of microarray data. Methods for narrowing the focus of the analysis by eliminating non-informative genes are helpful for reducing noise. One such method is the Significance of Microarray (SAM) algorithm proposed by Tibshirani's group at Stanford University (Tusher *et al.*, 2001). Current methods for filtering are not optimal and more work needs to be done in this area.

The SAM algorithm is permutation based and assumes that all of the genes are from the same population. The SAM analysis gives a list of genes which may be significant. Such a list could be used as a data filtering tool prior to cluster analysis. Genes having significance scores falling below a certain threshold could be treated as belonging to a single cluster and removed. Briefly, the steps in the SAM algorithm are,

1. Permute the data and compute test statistics (Equation 6.14) for each permutation.
2. Rank the test statistics in ascending order.
3. Compute mean test statistics for each ranking for all permutations.

4. Plot test statistic from the original ranking versus the mean test statistic from all of the permutations.
5. Define a distance from the mean permuted value to call significant.
6. Compute the false discovery rate (FDR) for this value.
7. Iterate until an "appropriate" FDR is obtained.

The FDR is calculated by dividing the median number of false positive genes for all of the permutations by the number of differentially expressed genes in the original data. As an example, the *t* test statistic is described in more detail. The algorithm essentially amounts to calculating the usual *t* statistic with a fudge factor as shown in Equation 6.14.

$$d_g = \frac{\overline{y}_{2g} - \overline{y}_{1g}}{\sqrt{\frac{1/n_1 + 1/n_2}{n_1 + n_2 - 2}\left(s_{1g}^2 + s_{2g}^2\right) + s_0^2}}, \tag{6.14}$$

where g is the gene number, $\overline{y}_{1g}$ and $\overline{y}_{2g}$ are the average $\log_2$ normalized hybridization values of gene g in the control and experimental strains, respectively, s_{1g}^2 and s_{2g}^2 are the variance estimates of gene g in the wild-type and experimental strains, respectively, n_1 and n_2 are the sample sizes for gene g in the wild-type and experimental strains, respectively, and s_0^2 is a small constant (or fudge factor) based on the median standard deviation for all genes (Tusher *et al.*, 2001). The purpose of the fudge factor is to correct for genes which have small denominators for d_g in order to avoid an inflation of significance. The SAM algorithm can miss genes with small changes in expression and does not easily adapt for more complex microarray designs. The t-test is robust for moderate sample sizes and hence, the SAM algorithm leads to an improvement especially when there are very few replicates in the design.

6.4 Discussion and Limitations

This dissertation delves into the application of clustering methods to the analysis of microarray data. The two dimensional normal mixture model is useful for producing clustering results. However, these results should be used in collaboration with the results from other methods before a biological result is claimed, as there is a danger of oversimplifying the biology.

An important issue which was not fully discussed in this dissertation is data normalization. There is disagreement in the literature over how to normalize microarray data. Clearly, there is a need to think about normalization, particularly if one is interested in comparing the results of multiple experiments or multiple arrays.

The clustering algorithms presented in Chapters 4 and 5 require the specification of starting values for the parameters. For the analyses, starting values were generated using the results of a k-means cluster analysis. The Expectation/Maximization (EM) algorithm is not sensitive to starting values (McLachlan and Peel, 2000). However, poor starting values often cause models to take much longer to converge. Due to numerical precision issues, it is possible for the EM algorithm to fail to converge. In our experience, this happens very rarely (< 1% of the time in the simulations). In such cases, new starting values were chosen and the model was run again.

Clustering microarray data is a difficult task due to the high degree of noise that is often present. The two dimensional clustering techniques presented in this dissertation may be more easily applied to other applications. Clustering may not be the preferred approach for the analysis of microarray data, but the parametric approach proposed in this dissertation is likely to be more useful in applications due to allowing an estimate of the probability of an observation belonging to any cluster.

Microarray data analyses assume that the gene expression measures reflect the underlying biology and are measured accurately. If the data is not reliable, even complex statistical modeling will not yield valid results. As microarray technology matures, it should be possible to obtain more accurate measures of gene expression, including experiments with a number of replicates. Once microarray analysis techniques become more standardized and computing hardware improves, it should become easier to perform and analyze replicated microarray experiments.

The methods proposed in this dissertation are limited by the assumption of normality and gene independence. These limitations are shared by most techniques for analyzing microarray data. If the data is non-normal, the clustering results may be biased and therefore normalizing transformation should be considered. Fortunately, there are tools available for evaluating the normality of data such as the Shapiro-Wilks test and normal quantile plots (Johnson and Wichern, 1998). It is possible to construct nearly any distribution from an appropriate mixture of normal distributions. The algorithms given in this dissertation could be adapted for mixtures of non-normal components. Gene independence is a huge assumption common to most methods for microarray data analysis. This assumption is typically made in order to make the models more computationally tractable. Relaxing this assumption would require some sort of multivariate analysis which estimated the covariance matrix for the genes. Replication aids in reducing noise and allows for a measure of variability. Replication also allows the relaxation of some of the stringent assumptions such as independence across genes.

Studying gene expression is only the first step in understanding the biological processes regulated by genes. Clustering based on the gene expression values is also preliminary. Gene pathways and protein expression are much more complex problems. Genes can code for multiple proteins and several genes can code for the same protein. The proteins act on a lower biological level than genes. Understanding the regulatory pathways and being able to selectively interrupt or modify protein expression is crucial to the development of genomic medicine.

Clearly, there is a continued need for new approaches to the analysis of microarray data. Bioinformatics is a young science and many open problems remain for the microarray research community. Opportunities exist in every phase of microarray research, from experimental design to final analysis.

Bibliography

Bibliography

Aitkin, M. & Aitkin, I. (1996). A Hybrid EM/Gauss-Newton Algorithm for Maximum Likelihood in Mixture Distributions. Statistics and Computing, 6, 127-130.

Akaike, H. (1974). A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control, 19, 716-723.

Allison, D., Gadbury, G., Heo, M., Fernandez, J., Lee, C., Prolla, T., & Weindruch, R. (2002). A Mixture Model Approach for the Analysis of Microarray Gene Expression Data. Computational Statistics and Data Analysis, 39, 1-20.

Bickel, P. J. & Doksum, K. A. (2001). Mathematical Statistics: Basic Ideas and Selected Topics. (2nd ed.) (Vols. 1) Upper Saddle River, New Jersey: Prentice Hall.

Bittner, M., Meltzer, P., Chen, Y., Jiangm Y., Seftor, E., Hendrix, M., Radmacher, M., Simon, R., Yakhini, Z., Ben-Dor, A., Sampas, N., Dougherty, E., Wang, E., Marincola, F., Gooden, C., Lueders, J., Glatfelter, A., Pollock, P., Carpten, J., Gillanders, E., Leja, D., Dietrich, K., Beaudry, C., Berens, M., Alberts, D., & Sondak, V. (2000). Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling. Nature, 406, 536-540.

Blashfield, R. K. (1976). Mixture Model Tests of Cluster Analysis: Accuracy of Four Agglomerative Hierarchical Methods. Psychological Bulletin, 83, 377-388.

Blashfield, R. K. & Aldenderfer, M. S. (1978). The Literature on Cluster Analysis. Multivariate Behavioral Research, 13, 271-295.

Bohr, H. (2003). Neural Network Prediction of Protein Structures. New York: Springer Verlag.

Bolstad, B., Irizarry, R., Astrand, M., & Speed, T. (2003). A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Variance and Bias. Bioinformatics, 19, 185-193.

Bozdogan, H. & Sclove, S. (1984). Multi-Sample Cluster Analysis Using Akaike's Information Criterion. Annals of the Institute of Statistic Mathematics, 36, 163-180.

Bradley, P. S., Fayyad, U. M., & Reina, C. A. (1999). Scaling EM Clustering to Large Databases. In Technical Report No. MSR-TR-98-35 (Seattle: Microsoft Research).

Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C., Gaasterland, T., Glenisson, P., Holstege, F. C., Kim, I. F., Markowitz, V., Matese, J. C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., & Vingron, M. (2001). Minimum Information about a Microarray Experiment (MIAME) - Toward Standards for Microarray Data. Nature Genetics, 29, 365-371.

Breckenridge, J. N. (1989). Replicating Cluster Analysis: Method, Consistency, and Validity. Multivariate Behavioral Research, 24, 147-161.

Brown, P. & Botstein, D. (1999). Exploring the New World of the Genome with DNA Microarrays. Nature Genetics, 21, 33-37.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., & Herskowitz, I. (1998). The Transcriptional Program of Sporulation in Budding Yeast. Science, 282, 699-705.

Cicchetti, D. V. & Allison, T. (1971). A New Procedure for Assessing Reliability of Scoring EEG Sleep Recordings. American Journal of EEG Technology, 11, 101-109.

Cleveland, W. S. & Grosse, E. (1991). Computational Methods for Local Regression. Statistics and Computing, 1, 47-62.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. Educational and Psychological Measurement, 20, 37-46.

Cowgill, M. (1993). Monte Carlo Validation of Two Genetic Clustering Algorithms. PhD Virginia Polytechnic Institute and State University.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society B, 39, 1-38.

Dhanasekaran, S., Barrette, T., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pienta, K., Rubin, M., & Chinnaiyan, A. (2001). Delineation of Prognostic Biomarkers in Prostate Cancer. Nature, 412, 822-826.

Eisen, M., Spellman, P., Brown, P., & Botstein, D. (1998). Cluster Analysis and Display of Genome-Wide Expression Patterns. Proceedings of the National Academy of Sciences, 95, 14863-14868.

Everitt, B. S. & Hand, D. J. (1981). Finite Mixture Distributions. London, England: Chapman and Hall CRC.

Everitt, B. S., Landau, S., & Leese, M. (2001). Cluster Analysis. (4th ed.) New York, NY: Oxford University Press, Inc.

Fleiss, J. L., Cohen, J., & Everitt, B. S. (1969). Large Sample Standard Errors of Kappa and Weighted Kappa. Psychological Bulletin, 72, 323-327.

Fleiss, J. L. & Cohen, J. (1973). The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability. Educational and Psychological Measurement, 33, 613-619.

Fowlkes, E. B. & Mallows, C. L. (1983). A Method for Comparing Two Hierachical Clustering Algorithms. Journal of the American Statistical Association, 78, 553-569.

Ghosh, D. & Chinnaiyan, A. (2002). Mixture Modeling of Gene Expression Data from Microarray Experiments. Bioinformatics, 18, 275-286.

Goldberg, D. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Boston, MA: Addison-Wesley.

Gower, J. C. & Legendre, P. (1986). Metric and Euclidean Properties of Dissimilarity Coefficients. Journal of Classification, 5, 5-48.

Gurney, K. (1997). An Introduction to Neural Networks. London: UCL Press.

Harvey, E. (2001). Predicting Stock Market Performance with Genetic Algorithms. M.S. Strayer University.

Hasselblad, V. (1966). Estimation of Parameters for a Mixture of Normal Distributions. Technometrics, 8, 431-446.

Heath, M. T. (1997). Scientific Computing: An Introductory Survey. Boston, Massachusetts: WCB/McGraw-Hill.

Herrero, J., Valencia, A., & Dopazo, J. (2001). A Hierarchical Unsupervised Growing Neural Network for Clustering Gene Expression Patterns. Bioinformatics, 17, 126-136.

Holland, J. (1975). Adaptation in Natural and Artificial Systems. Cambridge, MA: The MIT Press.

Hubert, L. J. & Arabie, P. (1985). Comparing Partitions. Journal of Classification, 2, 193-218.

Jaccard, P. (1908). Nouvelles Recherches Sur la Distribution Florale. Bulletin de la Societe Vaudoise de Sciences Naturelles, 44, 223-370.

Johnson, R. A. & Wichern, D. W. (1998). Applied Multivariate Statistical Analysis. (4th ed.) Upper Saddle River, NJ: Prentice-Hall, Inc.

Kaufman, L. & Rousseeuw, P. J. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. New York, New York: Wiley-Interscience.

Kerr, K. & Churchill, G. (2001). Statistical Design and the Analysis of Gene Expression Microarrays. Genetical Research, 77, 123-128.

Kerr, M. K. & Churchill, G. A. (2001). Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions from Microarray Experiments. PNAS, 98, 8961-8965.

Kuiper, F. K. & Fisher, L. (1975). A Monte Carlo Comparison of Six Clustering Procedures. Biometrics, 31, 777-783.

Lazarsfeld, P. F. (1950). The Logical and Mathematical Foundation of Latent Structure Analysis. In Studies in Social Psychology in World War II Volume IV (pp. 362-412).

Levine, R. (1981). Sex Differences in Schizophrenia: Timing or Subtypes? Psychological Bulletin, 90, 432-444.

Little, R. J. & Rubin, D. B. (1987). Statistical Analysis with Missing Data. New York, New York: Wiley.

Liu, G. L. (1968). Introduction to Combinatorial Mathematics. New York, New York: McGraw Hill.

Lockhart, D., Dong, H., Follettie, M., Gallo, M., Chee, M., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., & Brown, E. (1996). Expression Monitoring by Hybridization to High-Density Oligonucleotide Arrays. Nature Biotechnology, 14, 1675-1680.

Lockhart, D. (2002). Introduction to Microarrays. In Durham, N.C.: CAMDA Conference.

MacQueen, J. B. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of 5th Berkeley Symposium on Mathematical Sciences and Probability (1st ed., pp. 281-297). Berkeley, CA: University of California Press.

McLachlan, G. J. & Basford, K. E. (1988). Mixture Models: Inference and Applications to Clustering. New York, New York: Marcel Dekker.

McLachlan, G. J. & Krishnan, T. (1997). The EM Algorithm and Extensions. New York, New York: Wiley.

McLachlan, G. J. & Peel, D. (2000). Finite Mixture Models. New York, New York: John Wiley and Sons, Inc.

McLachlan, G. J., Bean, R., & Peel, D. (2002). A Mixture Model Based Approach to the Cluster of Microarray Expression Data. *Bioinformatics, 18,* 413-422.

Milligan, G. W. (1980). An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms. *Psychometrika, 45,* 325-342.

Milligan, G. W. & Cooper, M. C. (1986). A Study of the Comparability of External Criteria for Hierarchical Cluster Analysis. *Multivariate Behavioral Research, 21,* 41-58.

Mjolsness, E., Tobias, M., Castano, R., & Wold, B. (2000). From Coexpression to Coregulation: An Approach to Inferring Transcriptional Regulation Among Gene Classes from Large-Scale Expression Data. In S.Solla, T. Leen, & K. Muller (Eds.), *Advances in Neural Information Processing Systems* (pp. 928-934). Boston, MA: MIT Press.

Mojena, R. (1977). Hierarchical Grouping Methods and Stopping Rules: An Evaluation. *Computer Journal, 20,* 359-363.

Mujumdar, R., Ernst, L., Mujumdar, S., Lewis, C., & Waggoner, A. (1993). Cyanine Dye Labeling Reagents: Sulfoindocyanine Succinimidyl Esters. *Bioconjug Chem, 4,* 105-111.

Neal, R. M. & Hinton, G. E. (1998). A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants. In M.I.Jordan (Ed.), *Learning in Graphical Models* (pp. 355-368). Dordrecht: Kluwer.

Neale, M. & Cardon, L. (1992). *Methodology for Genetic Studies of Twins and Families.* Kluwer: Academic Press.

Neale, M. (2003). A Finite Mixture Distribution Model for Data Collected from Twins. *Twin Research, 6*.

Orchard, T. & Woodbury, M. A. (1972). A Missing Information Principle: Theory and Applications. *Proceedings of the 6th Symposium on Mathematical Statistics and Probability, 1,* 697-715.

Pan, W. (2002). A Comparative Review of Statistical Methods for Discovering Differentially Expressed Genes in Microarray Experiments. *Bioinformatics, 18,* 546-554.

Paull, K. D. (1989). Display and Analysis of Patterns of Differential Activity of Drugs Against Human Tumor Cell Lines: Development of Mean Graph and COMPARE Algorithm. *Journal of the National Cancer Institute, 81,* 1088-1092.

Pearson, K. (1894). Contribution to the Mathematical Theory of Evolution. *Philosophical Transactions A, 185,* 71-110.

Quackenbush, J. (2001). Computational Analysis of Microarray Data. Nature Reviews Genetics, 2, 418-427.

Quackenbush, J. (2002). Microarray Data Transformation and Normalization. Nature Reviews Genetics, 32, 496-501.

Rand, W. M. (1971). Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association, 66, 846-850.

Redner, R. A. & Walker, H. F. (1984). Mixture Densities, Maximum Likelihood, and the EM Algorithm. SIAM Review, 26, 159-239.

Rocke, D. M. & Durbin, B. (2001). A Model for Measurement Error for Gene Expression Arrays. Journal of Computational Biology, 8, 557-569.

Rogers, D. J. & Tanimoto, T. T. (1960). A Computer Program for Classifying Plants. Science, 132, 1115-1118.

Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S. S., vad de Rijn, M., Waltham, M., Pergamenschikov, A., Lee, J., Lashkari, D., Shalon, D., Myers, T., Weinstein, J., Botstein, D., & Brown, P. (2000). Systematic Variation in Gene Expression Patterns in Human Cancer Cell Lines. Nature Genetics, 24, 227-235.

Sagan, C. (1986). Contact. New York, New York: Pocket Books.

SAS Institute Inc. (1999). SAS/STAT User's Guide, Version 8. Cary, N.C.: SAS Institute Inc.

Schena, M., Shalon, D., Davis, R., & Brown, P. (1995). Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. Science, 270, 467-470.

Schwarz (1978). Estimating the Dimension of a Model. Annals of Statistics, 6, 461-464.

Sclove, S. (1987). Application of Model-Selection Criteria to Some Problems in Multivariate Analysis. Psychometrika, 52, 333-343.

Sneath, P. H. (1957). The Application of Computers to Taxonomy. Journal of General Microbiology, 17, 201-226.

Sokal, R. R. & Michener, C. D. (1958). A Statistical Method for Evaluating Systematic Relationships. University of Kansas Scientific Bulletin, 38, 1409-1438.

Sokal, R. R. & Sneath, P. H. A. (1963). Principles of Numerical Taxonomy. San Francisco, CA: W.H. Freeman.

Sorensen, T. (1948). A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons. Biol.Skr., 5, 1-34.

Staunton, J., Slonim, D., Coller, H., Tamayo, P., Angelo, M., Park, J., Scherf, U., Lee, J., Reinhold, W., Weinstein, J., Mesirov, J., Lander, E., & Golub, T. (2001). Chemosensitivity Prediction by Transcriptional Profiling. PNAS, 98, 10787-10792.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., & Golub, T. (1999). Interpreting Patterns of Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation. Proceedings of the National Academy of Sciences, 96, 2907-2912.

Tibshirani, R., Walther, G., & Hastie, T. (2000). Estimating the Number of Clusters in a Dataset via the Gap Statistic. Technical Report 208: Department of Statistics, Stanford University.

Titterington, D. M., Smith, A. F., & Makov, U. E. (1985). Statistical Analysis of Finite Mixture Distributions. New York, New York: Wiley.

Toronen, P., Kolehmainen, M., Wong, G., & Castren, E. (1999). Analysis of Gene Expression Data Using Self-Organizing Maps. FEBS Letters, 451, 142-146.

Tusher, V., Tibshirani, R., & Chu, G. Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. PNAS 98[9], 5116-5121. 2001.
Ref Type: Journal (Full)

van Osdol, V. W., Myers, T. G., Paull, K. D., Kohn, K. W., & Weinstein, J. N. (1994). Use of the Kohonen Self Organizing Map to Study the Mechanisms of Action of Chemotherapeutic Agents. Journal of the National Cancer Institute, 86, 1853-1859.

Wang, D., Ressom, H., Mussavi, M., & Domnisoru, C. (2002). Double Self-Organizing Maps to Cluster Gene Expression Data. Proceedings of the European Symposium on Artificial Neural Networks, 2002, 45-50.

Ward, J. H. (1963). Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association, 58, 236-244.

Ward, J. H. & Hook, M. E. (1963). Application of an Hierarchical Grouping Procedure to a Problem of Grouping Profiles. Educational and Psychological Measurement, 23, 69-82.

Weinstein, J. N. (1992). Neural Computing in Cancer Drug Development: Predicting Mechanism of Action. Science, 258, 447-451.

Weinstein, J. N. (1997). An Information-Sensitive Approach to the Molecular Pharmacology of Cancer. Science, 275, 343-349.

Weldon, W. (1892). Certain Correlated Variations in Crangon Vulgaris. Proceedings of the Royal Society of London, 51, 1-21.

Wishart, D. (1999). Clustan Graphics Primer: A Guide to Cluster Analysis. Edinburgh, Scotland: Clustan Limited.

Wodicka, L., Dong, H., Mittmann, M., Ho, M. H., & Lockhart, D. J. (1997). Genome Wide Expression Monitoring in Saccharomyces Cerevisiae. Nature Biotechnology, 15, 1359-1367.

Wolfinger, R. D., Gibson, G., Wolfinger, E. D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., & Paules, R. S. (2001). Assessing Gene Significance from cDNA Microarray Expression Data via Mixed Models. Journal of Computational Biology, 8, 625-637.

Woodbury, M. A. & Manton, K. G. (1982). A New Procedure for the Analysis of Medical Classification. Methods of Information in Medicine, 21, 210-220.

Woodbury, M. A., Manton, K. G., & Tolley, H. D. (1994). A General Model for Statistical Analysis Using Fuzzy Sets. Information Sciences, 1, 149-180.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A., & Ruzzo, W. (2001). Model Based Clustering and Data Transformation for Gene Expression Data. In Technical Report no.396 (Department of Statistics at the University of Washington).

Yeung, K. Y. & Ruzzo, W. (2001). Principal Component Analysis for Clustering Gene Expression Data. Bioinformatics, 17, 763-774.

Chapter 1 Appendices

1.1 SAS Code for Calculating Stirling Numbers

```
/* Calculates Stirling numbers of second kind based on the formula given by Johnson and
Wichern on page 727 */

proc iml;
        n     = 25;
        count = 0;

        do k = 3 to 3;
                temp = 0;
                do j = 0 to k;
                        temp = temp + ((-1)**(k-j) * comb(k,j) * (j**n));
                end;
                count = (1/fact(k))*temp;
        end;

        print count;

        quit;
        run;
```

Chapter 2 Appendices

2.1 SAS Code for Calculating LOESS Smoothed Data

```
options nodate;

libname lib 'c:\eric\clusterpaper\chu\log10ratioprofile\';

data work.temp;
        set lib.chuformatted2;
        ratio1 = log10(ratio1);
        ratio2 = log10(ratio2);
        ratio3 = log10(ratio3);
        ratio4 = log10(ratio4);
        ratio5 = log10(ratio5);
        ratio6 = log10(ratio6);
        ratio7 = log10(ratio7);

proc transpose data=work.temp out=work.chuformattedtransposed2;

data lib.chuformattedtransposed2;
        set work.chuformattedtransposed2;
        if _n_ = 1 then timepoint = 0;
        if _n_ = 2 then timepoint = 0.5;
        if _n_ = 3 then timepoint = 2;
        if _n_ = 4 then timepoint = 5;
        if _n_ = 5 then timepoint = 7;
        if _n_ = 6 then timepoint = 9;
        if _n_ = 7 then timepoint = 12;

%macro rungene(genenum);
        proc loess data=lib.chuformattedtransposed2;
                model col&genenum = timepoint /degree=2;
                ods output outputstatistics=work.results;

        %if &genenum=1 %then %do;
                data combined;
                        set work.results;
                        drop smoothingparameter obs depvar pred;
        %end;
        data temp;
                set work.results;
                drop smoothingparameter obs timepoint;

        data combined;
                merge combined temp;
                drop depvar;
                rename pred = pred&genenum;
%mend;

%macro doit(num);
        %do i = 1 %to &num;
                %rungene(&i);
        %end;
%mend;

ods listing close;
%doit(6118);
ods listing;

proc contents data=work.combined;

data lib.loessdata;
        set work.combined;

*proc print data=work.combined;
run;
```

2.2 SAS Code for Calculating LOESS Smoothed Profiles

```
/* LOESS Smoothed Ratio Profiles */

options nodate;

libname lib 'c:\eric\clusterpaper\chu\log10ratioprofile\';

%macro runprofile(profilenum);
        proc loess data=lib.profiles1;
                model col&profilenum = col8 /degree=2;
                ods output outputstatistics=work.results;

        %if &profilenum=1 %then %do;
        data combined;
            set work.results;
            drop smoothingparameter obs depvar pred;
    %end;

    data temp;
        set work.results;
        drop smoothingparameter obs col8;

    data combined;
        merge combined temp;
        drop depvar;
        rename pred = pred&profilenum;
%mend;

%macro doit(num);
    %do i = 1 %to &num;
        %runprofile(&i);
    %end;
%mend;

%doit(7);

data lib.profiles2;
    set work.combined;
    rename col8 = timepoint;

proc print data=lib.profiles2;

run;
```

2.3 SAS Code for Performing LOESS Smoothed Filtering

```
options nodate;

libname lib 'c:\eric\clusterpaper\chu\log10ratioprofile\';

data lib.loessdata2;
    set lib.loessdata;

data work.profiles2temp;
    set lib.profiles2;
```

```
        rename pred1=col1 pred2=col2 pred3=col3 pred4=col4 pred5=col5 pred6=col6
pred7=col7;

%macro rungene(genenum);
        data work.temp;
                set lib.loessdata2;
                drop timepoint;

        proc transpose data=work.temp out=work.temp2;

        data work.temp3;
                set work.temp2;
                drop _name_ _label_;
                if (_N_ ^= &genenum) then delete;

        proc transpose data=work.temp3 out=work.temp4;

        data work.temp5;
                set work.temp4;
                rename col1 = gene&genenum;
                drop _name_;

        data work.merged;
                merge work.temp5 work.profiles2temp;

        proc corr data=work.merged pearson outp=work.pearson noprint;

                var gene&genenum col1 - col7;

        data work.temp6;
                set work.pearson;
                drop _name_ _type_ col1-col7;
                if (_n_ < 5) then delete;

        data work.corr;
                set work.temp6;

        %if &genenum=1 %then %do;
                data combined;
                        set work.corr;
        %end;

        data work.combined;
                merge work.combined work.corr;
```

```
%mend;

%macro doit(num);
        %do i = 1 %to &num;
                %rungene(&i);
        %end;
%mend;

%doit(6118);

proc transpose data=work.combined out=work.combined2;

data lib.filtered2;
        set work.combined2;

proc print data=lib.filtered2;

run;
```

Chapter 3 Appendices

3.1 SAS Code for Comparing Clustering Methods

```
libname lib 'c:\eric\comparingclusters\';

filename clusterA 'c:\eric\comparingclusters\keuc.txt';
data work.clusterA;
        infile clusterA dlm='09'x;
        input id cluster;

filename clusterB 'c:\eric\comparingclusters\kp.txt';
data work.clusterB;
        infile clusterB dlm='09'x;
        input id cluster;

proc iml;
        start sortmat(mat,len);
                sorted = 0;
                do while(sorted = 0);
                        sorted = 1;
                        do j = 1 to len-1;
                                if mat[j,1] > mat[j+1,1] then
                                        do;
                                                temp       = mat[j,1];
                                                mat[j,1]   = mat[j+1,1];
                                                mat[j+1,1] = temp;
                                                temp       = mat[j,2];
                                                mat[j,2]   = mat[j+1,2];
                                                mat[j+1,2] = temp;
                                                sorted          = 0;
                                        end;
                        end;
                end;
                return(mat);
        finish sortmat;

        use work.clusterA;
        read all into clusterA;
        close work.clusterA;
```

```
use work.clusterB;
read all into clusterB;
close work.clusterB;

n  = 60;          /* number of items clustered */
nc = 9;           /* number of clusters        */

/* direct comparison taking cluster labels at face value */
naive = j(n,2,0);
do i = 1 to n;
        naive[i,1] = clusterA[i,2];
        naive[i,2] = clusterB[i,2];
end;
create work.naive from naive;
append from naive;

/* comparison based on rankings of cluster sizes */
sortA = j(nc,2,0);
do i = 1 to nc;
        sum = 0;
        do j = 1 to n;
                if clusterA[j,2] = i then sum = sum + 1;
        end;
        sortA[i,1] = sum;
        sortA[i,2] = i;
end;
sortA = sortmat(sortA,nc);

sortB = j(nc,2,0);
do i = 1 to nc;
        sum = 0;
        do j = 1 to n;
                if clusterB[j,2] = i then sum = sum + 1;
        end;
        sortB[i,1] = sum;
        sortB[i,2] = i;
end;
sortB = sortmat(sortB,nc);

sorted = naive;
do i = 1 to nc;
        do j = 1 to n;
                if (naive[j,1] = sortA[i,2]) then sorted[j,1] = i;
```

```
            if (naive[j,2] = sortB[i,2]) then sorted[j,2] = i;
        end;
    end;
    create work.sorted from sorted;
    append from sorted;

    /* best case scenario */
    agreemat = j(nc,nc,0);
    do i = 1 to nc;
        do j = 1 to nc;
            do k = 1 to n;
                if ((naive[k,1] = i) & (naive[k,2] = j)) then agreemat[i,j] =
agreemat[i,j] + 1;
            end;
        end;
    end;

    newclusters = j(nc,2,0);
    index = 1;
    do j = 1 to nc;
        big = -1;
        do i = 1 to nc;
            if (agreemat[i,j] > big) then
                do;
                    big      = agreemat[i,j];
                    xcluster = j;
                    ycluster = i;
                end;
        end;
        newclusters[index,1] = ycluster;
        newclusters[index,2] = xcluster;
        index = index + 1;
        do l = 1 to nc;
            agreemat[ycluster,l] = -1;
        end;
    end;

    bestcase = naive;
    do i = 1 to nc;
        do j = 1 to n;
            if (naive[j,1] = newclusters[i,1]) then bestcase[j,1] =
newclusters[i,2];
        end;
    end;
    create work.bestcase from bestcase;
    append from bestcase;

    agreemat2 = j(nc,nc,0);
    do i = 1 to nc;
        do j = 1 to nc;
            do k = 1 to n;
                if ((bestcase[k,1] = i) & (bestcase[k,2] = j)) then
agreemat2[i,j] = agreemat2[i,j] + 1;
            end;
        end;
    end;
    saveagreemat = agreemat2;

    /* Rand index */
    /*
    sample = {1 1 0, 1 2 1, 0 0 4};
    sample = {0 0 50, 2 47 0, 36 15 0};
    print sample;
    nc1  = 3;
    nc2  = 3;
    ntot = sample[+,+];
    */
    sample = saveagreemat;
    nc1    = nc;
    nc2    = nc;
    ntot   = sample[+,+];

    sum1 = comb(ntot,2);
    sum2 = 0;
    do i = 1 to nc1;
        do j = 1 to nc2;
            val  = sample[i,j];
            if (val >= 2) then sum2 = sum2 + comb(val,2);
        end;
    end;
    A = sum2;
    sum3 = 0;
    do i = 1 to nc1;
        val  = sample[i,+];
        if (val >= 2) then sum3 = sum3 + comb(val,2);
    end;
    B = sum3 - sum2;
    sum4 = 0;
```

```
do j = 1 to nc2;
        val  = sample[+,j];
        if (val >= 2) then sum4 = sum4 + comb(val,2);
end;
C = sum4 - sum2;
D = sum1 - A - B - C;
rand = (A + D) / sum1;
print rand;

/* adjusted Rand index */
/*
sample = {1 1 0, 1 2 1, 0 0 4};
sample = {0 0 50, 2 47 0, 36 15 0};
nc1  = 3;
nc2  = 3;
ntot = sample[+,+];
*/
sample = saveagreemat;
nc1    = nc;
nc2    = nc;
ntot   = sample[+,+];

sum1 = 0;
do i = 1 to nc1;
        do j = 1 to nc2;
                val = sample[i,j];
                if (val >= 2) then sum1 = sum1 + comb(val,2);
        end;
end;
sum2 = 0;
do i = 1 to nc1;
        val = sample[i,+];
        if (val >=2)then sum2 = sum2 + comb(val,2);
end;
sum3 = 0;
do j = 1 to nc2;
        val = sample[+,j];
        if (val >=2) then sum3 = sum3 + comb(val,2);
end;
bign = sum1  - (sum2*sum3)/comb(ntot,2);

sum4 = 0;
do i = 1 to nc1;
        val = sample[i,+];
```

```
                if (val >= 2) then sum4 = sum4 + comb(val,2);
        end;
        sum5 = 0;
        do j = 1 to nc2;
                val = sample[+,j];
                if (val >= 2) then sum5 = sum5 + comb(val,2);
        end;
        bigd = (sum4 + sum5)/2 - (sum2*sum3)/comb(ntot,2);
        adjrand = bign / bigd;
        print adjrand;

        print saveagreemat;
        quit;

/* naive comparison */
proc freq data=work.naive;
        tables col1*col2 / agree nopercent norow nocol nocum;

/* ranked comparison */
proc freq data=work.sorted;
        tables col1*col2 / agree nopercent norow nocol nocum;

/* best case comparison */
proc freq data=work.bestcase;
        tables col1*col2 / agree nopercent norow nocol nocum;

run;
```

Chapter 4 Appendices

4.1 Derivation of the EM Algorithm Formulas for the One Dimensional Normal Mixture Model

Notation:

y_i	= the observations indexed by i
C	= the number of mixture components
N	= the number of observations
α_{ik}	= the posterior probability of the i^{th} observation falling in the k^{th} group
π_k	= the proportion of observations falling in the k^{th} group
μ_k	= the k^{th} group mean
σ_k^2	= the k^{th} group variance.

Definitions:

The normal mixture model distribution:

$$f_{y_i}\left(y_i;\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma^2}\right)=\sum_{k=1}^{C}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)$$

The likelihood for the N observations:

$$L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma^2}\right)=\prod_{i=1}^{N}\sum_{k=1}^{C}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^2\right)$$

The log likelihood for the N observations with the restriction $\pi_C = 1-\sum_{k=1}^{C-1}\pi_k$:

$$\ell\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{2}\right)=\log\left[L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\mu},\ \boldsymbol{\sigma}^{2}\right)\right]$$

$$=\sum_{i=1}^{N}\log\left[\sum_{k=1}^{C}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^{2}\right)\right]$$

$$=\sum_{i=1}^{N}\log\left[\sum_{k=1}^{C-1}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_{\mathrm{k}}^{2}\right)+\left(1-\sum_{k=1}^{C-1}\pi_k\right)\phi\left(y_i;\ \mu_C,\ \sigma_{\mathrm{C}}^{2}\right)\right]$$

The posterior probability estimate, α_{ik}, is found by taking a weighted average of the C component densities. The formula for α_{ik} is:

$$\alpha_{ik}=\frac{\hat{\pi}_k\ \phi\left(y_i;\ \hat{\mu}_k,\ \hat{\sigma}_{\mathrm{k}}^{2}\right)}{\sum_{r=1}^{C}\hat{\pi}_r\ \phi\left(y_i;\ \hat{\mu}_r,\ \hat{\sigma}_{\mathrm{r}}^{2}\right)}$$

The estimates of the posterior probability, α_{ik}, are used to simplify the maximum likelihood estimates for the parameters. The MLE's are calculated in the usual way by taking the first derivatives of the log likelihood, setting them equal to zero, and solving. The derivations are shown below.

The estimate of π_k is found by taking the first derivative of ℓ with respect to π_k, setting it equal to 0, and solving the resulting equation for π_k under the constraint $\pi_C=1-\sum_{k=1}^{C-1}\pi_k$.

$$\frac{\partial\ell}{\partial\pi_k}=\sum_{i=1}^{N}\left[\frac{\phi\left(y_i;\ \mu_k,\ \sigma_k^{2}\right)\ -\ \phi\left(y_i;\ \mu_C,\ \sigma_C^{2}\right)}{\sum_{k=1}^{C}\pi_k\ \phi\left(y_i;\ \mu_k,\ \sigma_k^{2}\right)}\right]=0$$

Substituting for α_{ik} yields:

$$\sum_{i=1}^{N}\left(\frac{\alpha_{ik}}{\pi_k}-\frac{\alpha_{iC}}{\pi_C}\right)=\frac{\alpha_{.k}}{\pi_k}-\frac{\alpha_{.C}}{\pi_C}=\alpha_{.k}\left(1-\sum_{k=1}^{C-1}\pi_k\right)-\alpha_{.C}\pi_k=0$$

Adding up the C-1 equations and solving yields:

$$1-\pi_C = \frac{\sum_{k=1}^{C-1} \alpha_{.k}}{N}$$

This is equivalent to:

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} \hat{\alpha}_{ik}}{N}$$

The estimate of μ_k is found by taking the first derivative of ℓ with respect to μ_k, setting it equal to 0, and solving the resulting equation for μ_k.

$$\frac{\partial \ell}{\partial \mu_k} = \sum_{i=1}^{N} \left[\frac{\pi_k \ \phi'\left(y_i;\ \mu_k,\ \sigma_k^2\right)}{\sum_{k=1}^{C} \pi_k \ \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)} \right] = 0$$

$$\phi'\left(y_i;\ \mu_k,\ \sigma_k^2\right) = \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)\left(\frac{y_i-\mu_k}{\sigma_k^2}\right)$$

$$\sum_{i=1}^{N} \left[\frac{\pi_k \ \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)\left(\frac{y_i-\mu_k}{\sigma_k^2}\right)}{\sum_{k=1}^{C} \pi_k \ \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)} \right] = 0$$

Substituting for α_{ik} and solving yields:

$$\sum_{i=1}^{N} y_i \alpha_{ik} = \sum_{i=1}^{N} \mu_k \alpha_{ik}$$

Therefore,

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} y_i \hat{\alpha}_{ik}}{\sum_{i=1}^{N} \hat{\alpha}_{ik}}$$

The estimate of σ_k^2 is found by taking the first derivative of ℓ with respect to σ_k^2, setting it equal to 0, and solving the resulting equation for σ_k^2.

$$\frac{\partial \ell}{\partial \sigma_k^2} = \sum_{i=1}^{N} \left[\frac{\pi_k \ \phi'\left(y_i;\ \mu_k,\ \sigma_k^2\right)}{\sum_{k=1}^{C} \pi_k \ \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)} \right] = 0$$

$$\phi'\left(y_i;\ \mu_k,\ \sigma_k^2\right) = \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)\left(\frac{\left(y_i - \mu_k\right)^2}{\sigma_k^2} - 1\right)$$

$$\sum_{i=1}^{N} \left[\frac{\pi_k \ \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)\left(\frac{\left(y_i - \mu_k\right)^2}{\sigma_k^2} - 1\right)}{\sum_{k=1}^{C} \pi_k \ \phi\left(y_i;\ \mu_k,\ \sigma_k^2\right)} \right] = 0$$

Solving this equation yields:

$$\hat{\sigma}_k^2 = \frac{\sum_{i=1}^{N} \hat{\alpha}_{ik} \left(y_i - \hat{\mu}_k\right)^2}{\sum_{i=1}^{N} \hat{\alpha}_{ik}}$$

In the homogeneous variance case, σ^2 is found by first estimating all of the σ_k^2's and then taking a weighted average of these estimates.

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{C} \hat{\sigma}_k^2 \left(\sum_{i=1}^{N} \hat{\alpha}_{ik} \right)}{N}$$

4.2 SAS Code for Simulating a C-Component Normal Mixture Distribution

```
/* simulates a C component normal mixture distribution */

/* CHANGE THIS */
libname lib 'c:\eric\dissertation\chapter 4 supporting material\';

proc iml;
        seed   = 23112;                       /* seed for random number generator */

        /* CHANGE THIS */
        n      = 10000;                /* number of observations to be generated */
        nc     = 2;                           /* number of clusters*/

        p = j(nc,1,0);                 /* mixture proportions */
   mus = j(nc,1,0);             /* holds mus for mixture distribution */
        vars = j(nc,1,0);              /* holds variances for mixture distribution */
        sim = j(n,2,0);             /* holds simulated values (i, x) format */

        /* Simulation Parameters */
        /* CHANGE THESE */
        p[1,1] = 0.3;
        p[2,1] = 0.7;

        mus[1,1] = 5;
        mus[2,1] = 10;

        vars[1,1] = 2;
        vars[2,1] = 4;


        /* simulate the mixture data using appropriate proportions */
        do k = 1 to n;
                choice = ranuni(seed);
                left   = 0;
                right  = p[1,1];
                if (choice < right) then
```

```
            do;
                i = 1;
                mu    = mus[1,1];
                var   = vars[1,1];
                sig   = sqrt(var);
                x     = sig*rannor(seed) + mu;
                sim[k,1] = i;
                sim[k,2] = x;
            end;
        else
          do m = 2 to nc;
                left  = left  + p[m-1,1];
                right = right + p[m,1];
                if ((choice > left) & (choice < right)) then
                    do;
                        i = m;
                        mu    = mus[m,1];
                        var   = vars[m,1];
                        sig   = sqrt(var);
                        x     = sig*rannor(seed) + mu;
                        sim[k,1] = i;
                        sim[k,2] = x;
                    end;
            end;
    end;

    /* output parameters and simulated data */
    print p;
    print mus;
    print vars;

    /* CHANGE THIS */
    filename out 'c:\eric\dissertation\chapter 4 supporting material\simdata.dat';
file out;
    do i = 1 to n;
        put (sim[i,2]);
    end;
    closefile out;

    create lib.sim from sim;
    append from sim;

    quit;
```

```
proc print data=lib.sim;

run;
```

Chapter 5 Appendices

5.1 Derivation of the Conditional Mixture Distribution

Notation:

x_{ij} = the data point indexed by (i, j)

C = the number of mixture components for the rows

C^* = the number of mixture components for the columns

f = the mixture distribution for the rows

g = the mixture distribution for the columns

N = the number of rows of data

N^* = the number of columns of data

α_{ijkl} = the posterior probability of the $(i, j)^{th}$ observation falling in the $(k,l)^{th}$ group

π_k = the proportion of observations falling in the k^{th} group

ψ_l = the proportion of observations falling in the l^{th} group

μ_{kj} = the k^{th} group mean which changes with the column number, j

τ_{kl} = the $(k,l)^{th}$ group mean

σ_k = the k^{th} group standard deviation

$\sigma_{\tau l}$ = the l^{th} group standard deviation

The normal mixture distribution for the rows is:

$$f = \sum_{k=1}^{C} \pi_k f_{x_{ij}|\mu_{kj}} = \sum_{k=1}^{C} \pi_k \phi\left(y_{ij};\ \mu_{kj}, \sigma_k^2 \right).$$

The normal mixture distribution for the columns is:

$$g = \sum_{l=1}^{C^*} \psi_l f_{\mu_{kj}} = \sum_{l=1}^{C^*} \psi_l \phi\left(\mu_{kj};\ \tau_{kl}, \sigma_{\tau l}^2 \right).$$

The distribution of the conditional mixture is derived below.

$$
\begin{aligned}
f_{y_{ij}} &= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_k\psi_l \int_{-\infty}^{\infty} f_{x_{ij}|\mu_{kj}} f_{\mu_{kj}} \partial\mu_{kj} \\
&= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_k\psi_l \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}\left(x_{ij}-\mu_{kj}\right)^2 \right) \right] \text{x} \\
&\qquad \left[\frac{1}{\sqrt{2\pi}\sigma_{\tau l}} \exp\left(-\frac{1}{2\sigma_{\tau l}^2}\left(\mu_{kj}-\tau_{kl}\right)^2 \right) \right] \partial\mu_{kj} \\
&= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_k\psi_l \int_{-\infty}^{\infty} \left[\frac{1}{2\pi\sigma_k\sigma_{\tau l}} \exp\left(-\frac{1}{2\sigma_k^2}\left(x_{ij}-\mu_{kj}\right)^2 - \frac{1}{2\sigma_{\tau l}^2}\left(\mu_{kj}-\tau_{kl}\right)^2 \right) \right] \partial\mu_{kj} \\
&= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_k\psi_l \int_{-\infty}^{\infty} \left[\frac{1}{2\pi\sigma_k\sigma_{\tau l}} \exp\left(-\frac{1}{2\sigma_k^2}\left(x_{ij}^2 - 2x_{ij}\mu_{kj} + \mu_{kj}^2\right) - \right.\right. \\
&\qquad \left.\left. \frac{1}{2\sigma_{\tau l}^2}\left(\mu_{kj}^2 - 2\mu_{kj}\tau_{kl} + \tau_{kl}^2\right) \right) \right] \partial\mu_{kj} \\
&= \sum_{k=1}^{C}\sum_{l=1}^{C^*} \frac{\pi_k\psi_l}{2\pi\sigma_k\sigma_{\tau l}} \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2\sigma_k^2\sigma_{\tau l}^2}\left(x_{ij}^2\sigma_{\tau l}^2 - 2x_{ij}\mu_{kj}\sigma_{\tau l}^2 + \mu_{kj}^2\sigma_{\tau l}^2 + \right.\right. \\
&\qquad \left.\left. \mu_{kj}^2\sigma_k^2 - 2\mu_{kj}\tau_{kl}\sigma_k^2 + \tau_{kl}^2\sigma_k^2 \right) \right] \partial\mu_{kj} \\
&= \sum_{k=1}^{C}\sum_{l=1}^{C^*} \frac{\pi_k\psi_l}{2\pi\sigma_k\sigma_{\tau l}} \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2\sigma_k^2\sigma_{\tau l}^2}\left(\mu_{kj}^2\left(\sigma_k^2+\sigma_{\tau l}^2\right) - 2\mu_{kj}\left(x_{ij}\sigma_{\tau l}^2 + \tau_{kl}\sigma_k^2\right) + \right.\right. \\
&\qquad \left.\left. x_{ij}^2\sigma_{\tau l}^2 + \tau_{kl}^2\sigma_k^2 \right) \right] \partial\mu_{kj} \\
&= \sum_{k=1}^{C}\sum_{l=1}^{C^*} \frac{\pi_k\psi_l}{2\pi\sigma_k\sigma_{\tau l}} \exp\left[-\frac{1}{2\sigma_k^2\sigma_{\tau l}^2}\left(x_{ij}^2\sigma_{\tau l}^2 + \tau_{kl}^2\sigma_k^2 \right) \right] \text{x} \\
&\int_{-\infty}^{\infty} \exp\left[\frac{-\left(\sigma_k^2+\sigma_{\tau l}^2\right)}{2\sigma_k^2\sigma_{\tau l}^2} \left(\mu_{kj}^2 - \frac{2\mu_{kj}\left(x_{ij}\sigma_{\tau l}^2 + \tau_{kl}\sigma_k^2\right)}{\sigma_k^2+\sigma_{\tau l}^2} \right) \right] \partial\mu_{kj}
\end{aligned}
$$

$$= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\frac{\pi_k\psi_l}{2\pi\sigma_k\sigma_{\tau l}}\exp\left[-\frac{1}{2\sigma_k^2\sigma_{\tau l}^2}\left(x_{ij}^2\sigma_{\tau l}^2+\tau_{kl}^2\sigma_k^2-\frac{\left(\sigma_k^2+\sigma_{\tau l}^2\right)\left(x_{ij}\sigma_{\tau l}^2+\tau_{kl}\sigma_k^2\right)^2}{\left(\sigma_k^2+\sigma_{\tau l}^2\right)^2}\right)\right] \text{ x}$$

$$\int_{-\infty}^{\infty}\exp\left[\frac{-\left(\sigma_k^2+\sigma_{\tau l}^2\right)}{2\sigma_k^2\sigma_{\tau l}^2}\left(\mu_{kj}-\frac{x_{ij}\sigma_{\tau l}^2+\tau_{kl}\sigma_k^2}{\sigma_k^2+\sigma_{\tau l}^2}\right)^2\right]\partial\mu_{kj}$$

$$= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\left(\frac{\pi_k\psi_l}{\sqrt{2\pi}\sigma_k\sigma_\tau}\right)\left(\frac{\sigma_k\sigma_{\tau l}}{\sqrt{\sigma_k^2+\sigma_{\tau l}^2}}\right)\exp\left[-\frac{1}{2\sigma_k^2\sigma_{\tau l}^2}\left(x_{ij}^2\sigma_{\tau l}^2+\tau_{kl}^2\sigma_k^2-\right.\right.$$

$$\left.\left.\frac{\left(\sigma_k^2+\sigma_{\tau l}^2\right)\left(x_{ij}\sigma_{\tau l}^2+\tau_{kl}\sigma_k^2\right)^2}{\left(\sigma_k^2+\sigma_{\tau l}^2\right)^2}\right)\right] \text{ x}$$

$$\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}\left(\frac{\sqrt{\sigma_k^2+\sigma_{\tau l}^2}}{\sigma_k\sigma_{\tau l}}\right)\exp\left[\frac{-\left(\sigma_k^2+\sigma_{\tau l}^2\right)}{2\sigma_k^2\sigma_{\tau l}^2}\left(\mu_{kj}-\frac{x_{ij}\sigma_{\tau l}^2+\tau_{kl}\sigma_k^2}{\sigma_k^2+\sigma_{\tau l}^2}\right)^2\right]\partial\mu_{kj}$$

$$= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\left(\frac{\pi_k\psi_l}{\sqrt{2\pi}\sigma_k\sigma_{\tau l}}\right)\left(\frac{\sigma_k\sigma_{\tau l}}{\sqrt{\sigma_k^2+\sigma_{\tau l}^2}}\right)\exp\left[-\frac{1}{2\sigma_k^2\sigma_{\tau l}^2}\left(x_{ij}^2\sigma_{\tau l}^2+\tau_{kl}^2\sigma_k^2-\right.\right.$$

$$\left.\left.\frac{\left(\sigma_k^2+\sigma_{\tau l}^2\right)\left(x_{ij}\sigma_{\tau l}^2+\tau_{kl}\sigma_k^2\right)^2}{\left(\sigma_k^2+\sigma_{\tau l}^2\right)^2}\right)\right] \text{ x } \phi\left(\mu_{kj};\ \frac{x_{ij}\sigma_{\tau l}^2+\tau_{kl}\sigma_k^2}{\sigma_{\tau l}^2+\sigma_k^2},\ \frac{\sigma_k^2\sigma_{\tau l}^2}{\sigma_{\tau l}^2+\sigma_k^2}\right)$$

$$= \sum_{k=1}^{C}\sum_{l=1}^{C^*}\left[\frac{\pi_k\psi_l}{\sqrt{2\pi\left(\sigma_k^2+\sigma_{\tau l}^2\right)}}\right] \text{ x}$$

$$\exp\left[\frac{-\left(x_{ij}^2\sigma_{\tau l}^2\sigma_k^2+\cancel{\tau_{kl}^2\sigma_k^4}+\cancel{x_{ij}^2\sigma_{\tau l}^4}+\tau_{kl}^2\sigma_{\tau l}^2\sigma_k^2-\cancel{x_{ij}^2\sigma_{\tau l}^4}-2x_{ij}\tau_{kl}\sigma_{\tau l}^2\sigma_k^2-\cancel{\tau_{kl}^2\sigma_k^4}\right)}{2\sigma_k^2\sigma_{\tau l}^2\left(\sigma_k^2+\sigma_{\tau l}^2\right)}\right]$$

$$= \sum_{k=1}^{C} \sum_{l=1}^{C^*} \left[\frac{\pi_k \psi_l}{\sqrt{2\pi\left(\sigma_k^2 + \sigma_{\tau l}^2\right)}} \right] \exp\left[-\frac{1}{2\left(\sigma_k^2 + \sigma_{\tau l}^2\right)} \left(x_{ij}^2 - 2x_{ij}\tau_{kl} + \tau_{kl}^2 \right) \right]$$

$$= \sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_k \psi_l \phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_k^2 + \sigma_{\tau l}^2 \right)$$

$$= \sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl} \phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^2 \right), \text{ where } \pi_{\mathrm{kl}} = \pi_k \psi_l \text{ and } \sigma_{kl}^2 = \sigma_k^2 + \sigma_{\tau l}^2$$

5.2 Derivation of the EM Algorithm Formulas for the Two Dimensional Normal Mixture Model

Notation:

y_{ij} = the observations indexed by *i* (rows) and *j* (columns)

C = the number of mixture components across rows

C^* = the number of mixture components across columns

N = the number of rows of data

N^* = the number of columns of data

α_{ijkl} = the posterior probability of the $(i, j)^{th}$ observation falling in the $(k,l)^{th}$ group

π_{kl} = the proportion of observations falling in the $(k,l)^{th}$ group

τ_{kl} = the $(k,l)^{th}$ group mean

σ_{kl}^2 = the $(k,l)^{th}$ group variance.

Definitions:

The normal mixture model distribution:

$$f_{y_{ij}}\left(y_{ij};\ \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\sigma^2} \right) = \sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl} \phi\left(y_{ij};\ \tau_{\mathrm{kl}},\ \sigma_{\mathrm{kl}}^2 \right)$$

The likelihood for the NN^* observations:

$$L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)=\prod_{i=1}^{N}\prod_{j=1}^{N^*}\sum_{k=1}^{C}\sum_{l=1}^{C^*}\ \pi_{kl}\phi\left(y_{ij};\ \tau_{kl},\ \sigma_{\mathrm{kl}}^2\right)$$

The log likelihood for the NN^* observations with the global restriction $\pi_{CC^*}=1-\sum_{\substack{k=1\\kl\neq CC^*}}^{C}\sum_{l=1}^{C^*}\pi_{kl}$:

$$\begin{aligned}\ell\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)&=\log\left[L\left(\mathbf{y};\ \boldsymbol{\pi},\ \boldsymbol{\tau},\ \boldsymbol{\sigma}^{\mathbf{2}}\right)\right]\\&=\sum_{i=1}^{N}\sum_{j=1}^{N^*}\log\left[\sum_{k=1}^{C}\sum_{l=1}^{C^*}\ \pi_{kl}\phi\left(y_{ij};\ \tau_{kl},\ \sigma_{\mathrm{kl}}^2\right)\right]\\&=\sum_{i=1}^{N}\sum_{j=1}^{N^*}\log\left[\sum_{\substack{k=1\\kl\neq CC^*}}^{C}\sum_{l=1}^{C^*}\pi_{kl}\phi\left(y_{ij};\ \tau_{kl},\ \sigma_{\mathrm{kl}}^2\right)+\right.\\&\qquad\left.\left(1-\sum_{\substack{k=1\\kl\neq CC^*}}^{C}\sum_{l=1}^{C^*}\pi_{kl}\right)\phi\left(y_{ij};\ \tau_{CC^*},\ \sigma_{\mathrm{CC}^*}^2\right)\right]\end{aligned}$$

The posterior probability estimate, α_{ijkl}, is found by taking a weighted average of the CC^* component densities. The formula for α_{ijkl} is:

$$\alpha_{ijkl}=\frac{\hat{\pi}_{kl}\ \phi\left(y_{ij};\ \hat{\tau}_{kl},\ \hat{\sigma}_{\mathrm{kl}}^2\right)}{\sum_{r=1}^{C}\sum_{s=1}^{C^*}\hat{\pi}_{rs}\ \phi\left(y_{ij};\ \hat{\tau}_{rs},\ \hat{\sigma}_{\mathrm{rs}}^2\right)}$$

The estimates of the posterior probability, α_{ijkl}, are used to simplify the maximum likelihood estimates for the parameters. The MLE’s are calculated in the usual way by taking the first derivatives of the log likelihood, setting them equal to zero, and solving. The derivations are shown below.

The estimate of π_{kl} is found by taking the first derivative of ℓ with respect to π_{kl}, setting it equal to 0, and solving the resulting equation for π_{kl} under the constraint $\pi_{CC^*} = 1 - \sum_{\substack{k=1 \\ kl \neq CC^*}}^{C} \sum_{l=1}^{C^*} \pi_{kl}$.

$$\frac{\partial \ell}{\partial \pi_{kl}} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \left[\frac{\phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right) - \phi\left(y_{ij};\ \tau_{CC^*},\ \sigma_{CC^*}^2\right)}{\sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)} \right] = 0$$

Substituting for α_{ijkl} yields:

$$\sum_{i=1}^{N} \sum_{j=1}^{N^*} \left(\frac{\alpha_{ijkl}}{\pi_{kl}} - \frac{\alpha_{ijCC^*}}{\pi_{CC^*}} \right) = \frac{\alpha_{..kl}}{\pi_{kl}} - \frac{\alpha_{..CC^*}}{\pi_{CC^*}} = \alpha_{..kl} \left(1 - \sum_{\substack{k=1 \\ kl \neq CC^*}}^{C} \sum_{l=1}^{C^*} \pi_{kl} \right) - \alpha_{..CC^*} \pi_{kl} = 0$$

Adding up the $C(C^* - 1)$ equations and solving yields:

$$1 - \pi_{CC^*} = \frac{\sum_{\substack{k=1 \\ kl \neq CC^*}}^{C} \sum_{l=1}^{C^*} \alpha_{..kl}}{NN^*}$$

This is equivalent to:

$$\hat{\pi}_{kl} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl}}{NN^*}$$

The estimate of τ_{kl} is found by taking the first derivative of ℓ with respect to τ_{kl}, setting it equal to 0, and solving the resulting equation for τ_{kl}.

$$\frac{\partial \ell}{\partial \tau_{kl}} = \sum_{i=1}^{N}\sum_{j=1}^{N^*}\left[\frac{\pi_{kl}\ \phi'\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)}{\sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)}\right] = 0$$

$$\phi'\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right) = \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)\left(\frac{y_{ij}-\tau_{kl}}{\sigma_{kl}^2}\right)$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N^*}\left[\frac{\pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)\left(\frac{y_{ij}-\tau_{kl}}{\sigma_{kl}^2}\right)}{\sum_{k=1}^{C}\sum_{l=1}^{C^*}\pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)}\right] = 0$$

Substituting for α_{ijkl} and solving yields:

$$\sum_{i=1}^{N}\sum_{j=1}^{N^*} y_{ij}\alpha_{ijkl} = \sum_{i=1}^{N}\sum_{j=1}^{N^*}\tau_{kl}\alpha_{ijkl}$$

Therefore,

$$\hat{\tau}_{kl} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N^*} y_{ij}\hat{\alpha}_{ijkl}}{\sum_{i=1}^{N}\sum_{j=1}^{N^*}\hat{\alpha}_{ijkl}}$$

The estimate of σ_{kl}^2 is found by taking the first derivative of ℓ with respect to σ_{kl}^2, setting it equal to 0, and solving the resulting equation for σ_{kl}^2.

$$\frac{\partial \ell}{\partial \sigma_{kl}^2} = \sum_{i=1}^{N} \sum_{j=1}^{N^*} \left[\frac{\pi_{kl}\ \phi'\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)}{\sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)} \right] = 0$$

$$\phi'\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right) = \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right) \left(\frac{\left(y_{ij} - \tau_{kl}\right)^2}{\sigma_{kl}^2} - 1 \right)$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N^*} \left[\frac{\pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right) \left(\frac{\left(y_{ij} - \tau_{kl}\right)^2}{\sigma_{kl}^2} - 1 \right)}{\sum_{k=1}^{C} \sum_{l=1}^{C^*} \pi_{kl}\ \phi\left(y_{ij};\ \tau_{kl},\ \sigma_{kl}^2\right)} \right] = 0$$

Solving this equation yields:

$$\hat{\sigma}_{kl}^2 = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl} \left(y_{ij} - \hat{\tau}_{kl}\right)^2}{\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl}}$$

In the homogeneous variance case, σ^2 is found by first estimating all of the σ_{kl}^2's and then taking a weighted average of these estimates.

$$\hat{\sigma}^2 = \frac{\sum_{k=1}^{C} \sum_{l=1}^{C^*} \hat{\sigma}_{kl}^2 \left(\sum_{i=1}^{N} \sum_{j=1}^{N^*} \hat{\alpha}_{ijkl} \right)}{NN^*}$$

5.3 Derivatives for the Calculation of Two Dimensional Posterior Probability Confidence Intervals

The derivative of $\widehat{\alpha}_{ijkl}$ with respect to π_{kl} is:

$$\frac{\partial \widehat{\alpha}_{ijkl}}{\partial \pi_{kl}} = \frac{\partial}{\partial \pi_{kl}} \left[\frac{\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^{2}\right)}{\sum_{r=1}^{C}\sum_{s=1}^{C^{*}} \pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}}, \sigma_{rs}^{2}\right)} \right]$$

$$= \frac{\phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^{2}\right)\sum_{r=1}^{C}\sum_{s=1}^{C^{*}} \pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}}, \sigma_{rs}^{2}\right) - \pi_{kl}\left[\phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^{2}\right)\right]^{2}}{\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}} \pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}}, \sigma_{rs}^{2}\right)\right]^{2}}$$

$$= \frac{\phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^{2}\right)\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}} \pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}}, \sigma_{rs}^{2}\right) - \pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}}, \sigma_{kl}^{2}\right)\right]}{\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}} \pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}}, \sigma_{rs}^{2}\right)\right]^{2}}.$$

The derivative of $\widehat{\alpha}_{ijkl}$ with respect to τ_{kl} is:

$$
\begin{aligned}
\frac{\partial \widehat{\alpha}_{ijkl}}{\partial \tau_{kl}} &= \frac{\partial}{\partial \tau_{kl}}\left[\frac{\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)}{\sum\limits_{r=1}^{C}\sum\limits_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)}\right] \\
&= \left\{\pi_{kl}\left(y_{ij}-\tau_{kl}\right)\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)-\right. \\
&\qquad \left.\pi_{kl}^{2}\left(y_{ij}-\tau_{kl}\right)\left[\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\right]^{2}\right\}\ /\ \left\{\sigma_{kl}^{4}\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right]^{2}\right\} \\
&= \left\{\pi_{kl}\left(y_{ij}-\tau_{kl}\right)\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)-\right.\right. \\
&\qquad \left.\left.\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)\right]\right\}\ /\ \left\{\sigma_{kl}^{4}\left[\sum_{r=1}^{C}\sum_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right]^{2}\right\}.
\end{aligned}
$$

The derivative of $\widehat{\alpha}_{ijkl}$ with respect to σ_{kl} is:

$$\frac{\partial \hat{\alpha}_{ijkl}}{\partial \sigma_{kl}} = \frac{\partial}{\partial \sigma_{kl}} \left[\frac{\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)}{\sum\limits_{r=1}^{C}\sum\limits_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)} \right]$$

$$= \left\{ \left[\frac{\left(y_{ij}-\tau_{kl}\right)^{2}}{\sigma_{kl}^{2}} - 1 \right] \left[\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right) \sum\limits_{r=1}^{C}\sum\limits_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right) - \right.\right.$$

$$\left.\left. \pi_{kl}^{2}\left[\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right]^{2} \right] \right\} \Big/ \left\{ \sigma_{kl} \left[\sum\limits_{r=1}^{C}\sum\limits_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right) \right]^{2} \right\}$$

$$= \left[\frac{\pi_{kl}\phi\left(y_{ij};\ \tau_{\mathrm{kl}},\sigma_{kl}^{2}\right)}{\sigma_{kl}} \right] \left[\frac{\left(y_{ij}-\tau_{kl}\right)^{2}}{\sigma_{kl}^{2}} - 1 \right] \left[\left\{ \sum\limits_{r=1}^{C}\sum\limits_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right) - \right.\right.$$

$$\left.\left. \pi_{kl}\left[\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right)\right] \right\} \Big/ \left\{ \left[\sum\limits_{r=1}^{C}\sum\limits_{s=1}^{C^{*}}\pi_{rs}\phi\left(y_{ij};\ \tau_{\mathrm{rs}},\sigma_{rs}^{2}\right) \right]^{2} \right\} \right].$$

5.4 SAS Code for Simulating a Two Dimensional Normal Mixture Distribution

```
/* simulates 2 dimensional normal mixture distribution */

/* CHANGE THE LIBRARY PATH */
libname lib  'C:\eric\dissertation\chapter 5 supporting material\4 x 5 simulation\';
proc iml;
        seed  = 2003;              /* seed for random number generator */
        n     = 1000;              /* number of observations to be generated */
        nc1   = 4;                 /* number of clusters in dimension 1 */
        nc2   = 5;                 /* number of clusters in dimension 2 */
        nsim  = 1;                 /* number of simulations - set to 1 for single run */

        p = j(nc1,1,0);            /* dimension 1 mixture proportions */
        q = j(nc2,1,0);            /* dimension 2 mixture proportions */
        pq = j(nc1,nc2,0);         /* 2 dimensional mixture proportions */
```

```
    taus = j(nc1,nc2,0);    /* taus for 2 dimensional mixture component distributions
*/
    sigs = j(nc1,nc2,0);    /* sigmas for 2 dimensional mixture component
distributions */
    sim = j(n*nsim,6,0);    /* holds simulated values (i, j, x) format */

    /* Simulation Parameters */
    /* CHANGE THESE */

    start = 5;
    do i = 1 to nc1;
        do j = 1 to nc2;
            taus[i,j] = start;
            start = start + 5;
        end;
    end;

    start = 1;
    do i = 1 to nc1;
        do j = 1 to nc2;
            sigs[i,j] = sqrt(start);
            start = start + 1;
        end;
    end;

    p[1,1] = 0.6;
    p[2,1] = 0.1;
    p[3,1] = 0.15;
    p[4,1] = 0.15;

    q[1,1] = 0.4;
    q[2,1] = 0.2;
    q[3,1] = 0.1;
    q[4,1] = 0.15;
    q[5,1] = 0.15;

    /* Calculate 2 dimensional mixture proportions */
    props = j(nc1*nc2,3,0); /* i, j, p */
    index = 1;
    do i = 1 to nc1;
        do j = 1 to nc2;
            pq[i,j] = p[i,1]*q[j,1];
            props[index,1] = i;
            props[index,2] = j;
            props[index,3] = pq[i,j];
            index = index + 1;
        end;
    end;

    /* simulate the 2 dimensional mixture data using appropriate proportions */
    k = 1;
    do simnum = 1 to nsim;
    do a = 1 to 10;
        do b = 1 to 100;

        choice = ranuni(seed);
        left   = 0;
        right  = props[1,3];
        if (choice < right) then
            do;
                i = props[1,1];
                j = props[1,2];
                tau    = taus[i,j];
                sig    = sigs[i,j];
                x        = sig*rannor(seed) + tau;  /* f(xij|mu) = N(mu,sig) */
                sim[k,1] = i;
                sim[k,2] = j;
                sim[k,3] = x;
                sim[k,4] = 1;
                sim[k,5] = a;
                sim[k,6] = b;
            end;
        else
            do m = 2 to (nc1*nc2);
                left  = left  + props[m-1,3];
                right = right + props[m,3];
                if ((choice > left) & (choice < right)) then
                    do;
                        i = props[m,1];
                        j = props[m,2];
                        tau    = taus[i,j];
                        sig    = sigs[i,j];
                        x        = sig*rannor(seed) + tau;  /* f(xij|mu)
= N(mu,sig) */
                        sim[k,1] = i;
                        sim[k,2] = j;
                        sim[k,3] = x;
                        sim[k,4] = m;
                        sim[k,5] = a;
                        sim[k,6] = b;
                    end;
            end;
            k = k + 1;
        end;
    end;
    end;

    /* output parameters and simulated data */
    print props;
    print taus;
    print sigs;

    /* calculate and output expected value */
    expectval = 0;
    do i = 1 to nc1;
        do j = 1 to nc2;
            expectval = expectval + pq[i,j]*taus[i,j];
        end;
    end;
    print expectval;
    expectvar = 0;
    do i = 1 to nc1;
        do j = 1 to nc2;
            expectvar = expectvar + pq[i,j]*(taus[i,j]**2 + sigs[i,j]**2);
        end;
    end;
    expectvar = expectvar - expectval**2;
    print expectvar;

    /* CHANGE THE FILE PATH */
    filename out 'C:\eric\dissertation\chapter 5 supporting material\4 x 5
simulation\simdata.dat';
  file out;
    do i = 1 to n;
        put (sim[i,3]);
    end;
    closefile out;

    create lib.sim from sim;
    append from sim;
quit;
run;
```

5.5 List of the 429 Genes Selected from Ross *et al.* (2000) Data Set

AA011608	AA019718	AA019818	AA026911	AA033566	AA040702
AA044444	AA052987	AA053379	AA053413	AA053629	AA055477
AA055794	AA057618	AA057760	AA058541	H77542	N55093
N93414	W05361	W37533	W46396	W80489	W93388
AA001722	AA019164	AA025473	AA032095	AA034315	AA035384
AA035528	AA044439	AA053637	AA057384	H68937	H88060
N36777	N59231	N67639	N90435	N98611	W81517
W90021	AA018766	AA034050	AA044444	AA053413	AA053629
AA058541	H67775	H75535	N63143	N95053	R38619
R54011	R79944	W37533	W90417	W90633	AA007625
AA025547	AA026911	AA027090	AA031697	AA040872	AA043211
AA055481	AA057196	AA058529	N29759	N95169	T40987
W74235	W89012	W93355	AA031375	H01224	N23665
N65943	W73070	AA004674	AA026921	AA029558	AA034221
AA035404	AA037369	AA037371	AA037770	AA040875	AA040928
AA043188	AA043745	AA044596	AA045330	AA045811	AA045978
AA046035	AA046423	AA047247	AA054290	AA055764	H73006
N92990	W73368	W90071	AA029558	AA031478	AA034221
AA035255	AA035404	AA037788	AA044100	AA046423	AA046701
AA047042	AA047247	AA055644	AA055764	N92990	W73368
AA025679	AA037689	AA053546	AA055467	AA055540	N35315
R94927	W69491	W80357	AA026222	AA028001	AA028094
AA031493	AA035372	AA035645	AA037353	AA045032	AA045090
AA046063	AA046312	AA047261	AA047372	AA053076	AA053558
AA055077	AA055408	AA056738	H85095	H95849	N20225
N63511	N70518	N72074	N78838	N80723	N93479
R52703	R59373	W44416	W46479	W79419	W86907
AA007652	AA009800	AA019922	AA025243	AA029097	AA035619
AA042879	AA044930	AA046316	AA047408	AA056232	H17086
H68549	N90435	N99964	W02680	W80458	AA004292
AA004931	AA005215	AA025275	AA029438	AA031831	AA031961
AA034172	AA035186	AA035508	AA039344	AA039599	AA040161
AA040777	AA040878	AA041299	AA043504	AA043755	AA044077
AA046521	AA047268	AA047324	AA047462	AA047679	AA052906
AA053225	AA056204	AA056470	AA057359	AA057728	AA058523
AA058997	H12764	H26976	H52664	N27930	N36901
N48061	N58770	N58838	N66100	N66376	N67168
N68633	N72265	N79519	N95301	N98775	R06755
R11631	R16808	R42435	R43524	R90842	T40568
T51244	T95619	W23729	W37431	W42423	W42587

W47128	W49593	W69354	W69649	W73035	W74500
W80587	W84728	W90762	W92696	W93379	AA010417
AA035619	N51716	N54791	N74888	R48613	W85726
AA046018	AA053461	AA056394	W69491	W80357	AA010561
AA025594	AA025800	AA027277	AA032074	AA032096	AA033563
AA033850	AA035236	AA040535	AA045480	AA045739	AA045881
AA046047	AA046709	AA052985	AA053077	AA053116	AA053169
AA053308	AA053681	AA055732	AA056469	AA056611	AA057535
AA057546	AA057604	AA057638	AA057670	AA057732	AA057821
AA057826	AA058319	AA058350	AA058366	AA058368	AA284246
H16800	H30650	H40135	N33200	N64544	N94490
R51086	W15407	W72115	W87396	W95362	W95649
AA028001	AA031493	AA035372	AA045032	AA047372	AA053558
H85095	N20225	N72074	N78838	N80723	W46479
W79419	AA017566	AA031701	AA039330	N53668	N73082
W94196	AA029560	AA029723	AA031265	AA035579	AA042981
AA043237	AA045877	AA052950	AA053371	H61739	N62998
W05368	W73690	W73776	W74535	W90478	AA026089
AA034172	AA040878	AA044233	AA047324	AA058523	N66100
R16808	R20750	W37431	W49593	W78128	W93802
W96210	AA001916	AA026796	AA029438	AA039640	AA040617
AA045192	AA045874	AA055196	AA055664	AA059306	H14343
H50438	H52664	H56186	H59260	H62385	H65189
N30606	N66521	N72115	N74420	N90191	N94440
N94499	R73421	R80235	R83277	W42414	W42423
W73785	W74500	W80586	W86050	W95001	W95471
AA026207	AA031671	AA034172	AA035186	AA036724	AA040878
AA044233	AA046521	AA047243	AA053271	AA054226	AA056204
AA057736	AA058523	H23728	N77814	N95176	R11631
R16808	W45726	W69649	W92047	W93802	AA022690
AA041299	AA054312	AA056468	H22978	H29200	N49221
R44253	W04723	W92696			

5.6 173 Observation Zero Variance Cluster

Gene	Cell	Gene	Cell	Gene	Cell	Gene	Cell
AA011608	20	AA037369	57	AA043755	20	N20225	7
AA026911	26	AA037371	37	AA047268	28	W46479	20
AA052987	6	AA043745	32	AA047268	45	AA031701	5
W05361	5	AA055764	4	AA053225	42	AA031701	28
W80489	16	H73006	29	AA056204	28	AA029560	56
W80489	24	N92990	29	AA057728	34	AA029723	56
W80489	36	W73368	29	N66100	2	AA031265	56
AA001722	27	AA035255	7	N72265	13	AA053371	44
AA032095	18	AA035255	54	R11631	31	W73690	54
AA035528	18	AA047042	33	R16808	31	AA026089	15
AA044439	18	AA047247	47	R42435	31	AA034172	20
AA057384	28	W73368	8	R90842	43	N66100	56
H68937	49	AA028001	9	T40568	43	R16808	48
H88060	38	AA028001	19	T51244	13	W37431	46
N36777	38	AA028094	58	W23729	29	W78128	33
N90435	12	AA035372	19	W37431	20	AA039640	34
N98611	12	AA045090	14	W37431	40	AA045874	22
AA018766	7	AA047261	24	W42587	15	AA059306	7
AA018766	8	AA053076	19	W47128	18	H56186	13
AA018766	19	AA053558	19	W47128	40	H59260	13
H75535	60	H85095	50	W73035	38	H62385	59
N63143	4	N72074	52	W84728	18	N74420	58
R38619	16	R52703	55	N54791	29	N94499	19
W90417	53	R59373	55	AA027277	54	N94499	30
AA025547	37	W44416	27	AA032074	54	N94499	42
AA026911	7	W46479	35	AA040535	2	R73421	19
AA026911	35	AA019922	53	AA045881	38	R73421	30
AA043211	25	AA035619	24	AA046709	14	R73421	42
AA043211	41	AA035619	53	AA052985	5	AA040878	3
AA055481	25	AA042879	23	AA053116	47	AA040878	32
AA055481	41	AA044930	10	AA056469	9	AA047243	40
AA057196	52	AA044930	46	AA057535	59	AA057736	15
AA058529	19	AA044930	60	AA057604	59	H23728	54
N29759	19	AA046316	28	AA057638	9	N77814	15
T40987	26	AA056232	16	AA057638	47	N77814	32
T40987	32	AA056232	44	AA057670	9	N95176	20
W74235	10	AA056232	58	AA057670	47	R11631	20
W89012	10	H68549	53	AA057821	49	AA022690	12
AA031375	31	N90435	53	AA058350	42	N20225	7
H01224	31	W80458	39	H30650	27		
N23665	59	AA004292	39	N33200	15		

AA004674	8	AA031831	40	N94490	41
AA026921	8	AA034172	2	W87396	21
AA035404	13	AA039599	11	W95362	21
AA035404	44	AA041299	29	N20225	2

2-DCluster Manual

Software for Fitting Two Dimensional Normal Mixture Data

Version 1.0

June 2003

Eric Harvey

eharvey@mail2.vcu.edu

Department of Biostatistics

Virginia Commonwealth University

Richmond, Virginia

This program was written for partial fulfillment of my dissertation requirements for the degree of Doctor of Philosophy in the Biostatistics Department at Virginia Commonwealth University.

Although every effort has gone into making 2-DCluster bug free, no warranty is expressed or implied. Use 2-DCluster at your own risk. The author can only provide limited support.

2-d.1 Introduction

2-DCluster was written to support the fitting of two dimensional normal mixture models, although one dimensional normal mixture models are also supported. The two dimensional mixture model is developed in Eric Harvey's dissertation available electronically at http://etd.vcu.edu. Two dimensional mixture models occur when the data can be represented in a grid with categorical labels across both axes. Clustering such data in one dimension requires the data to be collapsed across the opposite dimension. This data reduction results in a loss of information. Two dimensional mixture models use both dimensions of the data simultaneously to estimate the parameter values. 2-DCluster was developed on a Microsoft Windows XP machine and has been tested on Windows 2000 and Windows Me machines. Microsoft Visual Basic version 6.0 and Lahey Fortran 95 version 5.7 were used to develop 2-DCluster. The source code is available at http://etd.vcu.edu and requires the Wisk library (www.lahey.com) to compile. 2-DCluster requires at least 5 megabytes of RAM to run, and more memory is required for large data sets. Every effort was made to make 2-DCluster as efficient as possible.

2-d.2 Algorithm Description

Maximum likelihood based methods are used to estimate the two dimensional normal mixture model parameters. The Expectation/Maximization algorithm is an iterative method used for estimation. Each cluster requires the estimation of a proportion, a mean, and a variance. Each observation has an estimated posterior probability of belonging to every cluster. A hybrid method which uses both the Netwon Raphson and Expectation/Maximization algorithms is also available. This algorithm was proposed by

Aitkin and Aitkin (1996). For two dimensional clustering, each observation is indexed by a row number and a column number. Further details on the algorithm are given in Eric Harvey's dissertation available electronically at http://etd.vcu.edu.

2-d.3 Program Installation

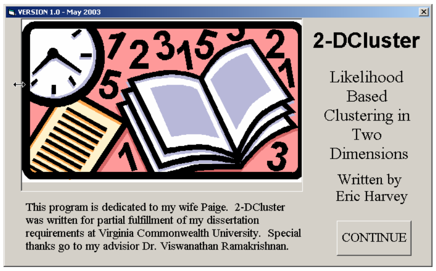
2-DCluster is distributed in one of two ways. For the CD's included with the bound dissertation copies, the 2-DCluster installation is the same as for a regular windows program. Insert the CD and run setup.exe. Follow the prompts to choose an installation directory and program group name. Run 2-DCluster by clicking the 2-DCluster icon in the 2-DCluster program group.

For downloaded copies of 2-DCluster (available at http://etd.vcu.edu), the installation files are contained in a self extracting archive file called 2dCinstall.exe. Copy this file to a temporary directory and run it. You will be prompted for a directory name to extract the files to. You may choose any directory you like. Once the files are extracted, change directories to the one which the files were extracted to. Double click on the file setup.exe to install 2-DCluster. You will be prompted for the installation directory for the program. Follow the operating system prompts. You will be informed when installation is complete.

NOTE: You may be prompted to update your system files. If this occurs, please cancel the setup and run Windows Update (accessible from Microsoft Internet Explorer) to update your system files. After this is complete, re-run the 2-DCluster setup program. The 2-DCluster distribution includes several example data files, as well as a documentation file named 2dcluster.pdf. An uninstall option is available in the 2-DCluster program group. 2-DCluster requires the availability of Adobe Acrobat Reader and Microsoft Internet Explorer for live display of the documentation. The export data feature requires Microsoft Excel to be installed.

2-d.4 User Interface Description

2-DCluster has a graphical user interface from which various parameters can be controlled. The program is run by double clicking on the 2dcluster icon. The startup screen is shown below. Click Continue to proceed to the next screen.



VERSION 1.0 - May 2003

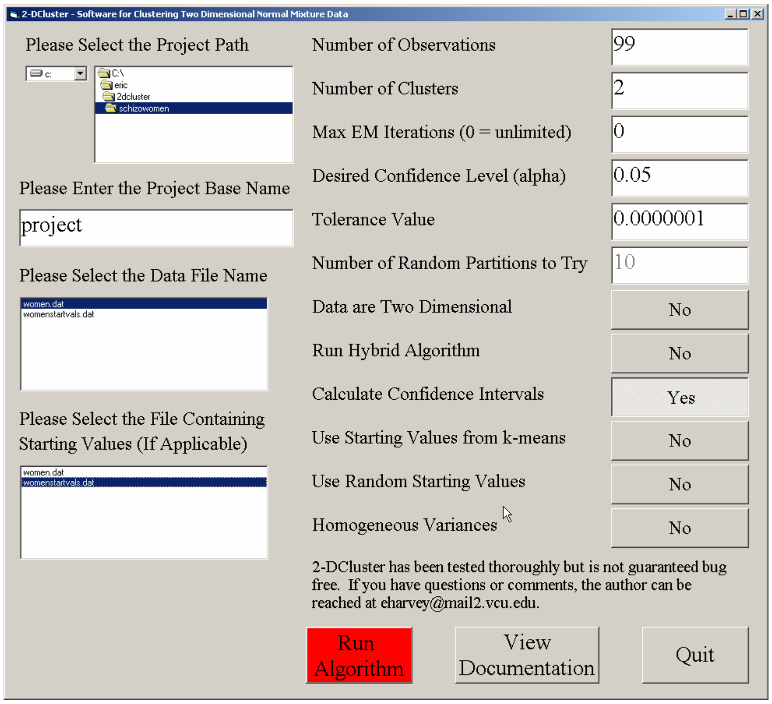
2-DCluster

Likelihood Based Clustering in Two Dimensions

Written by Eric Harvey

This program is dedicated to my wife Paige. 2-DCluster was written for partial fulfillment of my dissertation requirements at Virginia Commonwealth University. Special thanks go to my advisior Dr. Viswanathan Ramakrishnan.

The main screen is shown below and allows the user to control a variety of settings.



2-DCluster - Software for Clustering Two Dimensional Normal Mixture Data

Please Select the Project Path

c:

C:\
eric
2dcluster
schizowomen

Please Enter the Project Base Name

project

Please Select the Data File Name

women.dat
womenstartvals.dat

Please Select the File Containing Starting Values (If Applicable)

women.dat
womenstartvals.dat

Number of Observations: 99

Number of Clusters: 2

Max EM Iterations (0 = unlimited): 0

Desired Confidence Level (alpha): 0.05

Tolerance Value: 0.0000001

Number of Random Partitions to Try: 10

Data are Two Dimensional: No

Run Hybrid Algorithm: No

Calculate Confidence Intervals: Yes

Use Starting Values from k-means: No

Use Random Starting Values: No

Homogeneous Variances: No

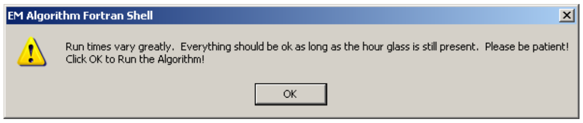
2-DCluster has been tested thoroughly but is not guaranteed bug free. If you have questions or comments, the author can be reached at eharvey@mail2.vcu.edu.

Each of the options on the control screen shown above is now discussed. The left hand column of options indicates where the data is located and where the files should be stored. Note that filenames containing the "_" character are not allowed. The project path indicates the directory that you wish to store the project in. The default directory is the directory in which 2-DCluster was installed. The project base name indicates the project name. Data files generated by 2-DCluster will be named *projectbasename.*EXT, where EXT represents the file extensions described below. The project base name defaults to PROJECT. The data file name indicates the file in which the data to be analyzed is stored. The format of the data file is described in Section 2-d.5. If no filename is specified, it defaults to DATA.DAT in the project directory. The starting value file indicates the file containing starting values for the algorithm. This box will not appear if the user requests k-means cluster or random starting values to be generated by 2-DCluster. The format of the start value file is described in Section 2-d.5. If no filename is specified, it defaults to STARTVALS.DAT.

Options in the right hand column allow algorithm parameters to be controlled. The number of observations indicates the number of data points contained in the data file. For program optimization, the number of observations is limited to 1,000,000. If your application has more observations than this, the program can be recompiled with a higher limit. The observation number corresponds to the row number in the data file from which the observation came. The number of clusters must be pre-specified. The default is 2 clusters. The number of clusters per dimension is limited to 500, which should be adequate for most applications. The maximum number of EM algorithm iterations may be set. If this is 0, the number of iterations is unlimited. The desired confidence level, alpha, indicates the alpha level used in calculating the confidence intervals. The default confidence level is 0.05 which yields 95% confidence intervals. This option is not available if confidence interval calculations are not requested. The tolerance level may be chosen based on the required accuracy of the application. The default value is 0.0000001 and is suitable for most applications. Smaller tolerance values may be chosen, but may negatively affect the convergence rate. If random starting values are requested, the number of random partitions to try may be specified. The default value is 10 partitions. 2-DCluster selects the random partition having the minimal within cluster variance.

The selection boxes may be changed by clicking on them. The first selection box indicates whether the data are two dimensional. Two dimensional data have a different file format, as discussed in Section 2-d.5. The default value of this setting is YES. The hybrid algorithm, as described by Aitkin and Aitkin (1996) , may be selected. The default value of this is NO, which indicates that the ordinary EM algorithm is used. Confidence intervals may be requested. The default value for this is YES. Starting values based on a k-means cluster analysis may be generated by 2-DCluster. The default value for this is NO. Starting values may also be randomly generated. The default value for this is NO. Only one method of generating starting values may be chosen at a time. Finally, the variance estimates may be forced to be homogeneous if appropriate for the application. The default value for this is NO.

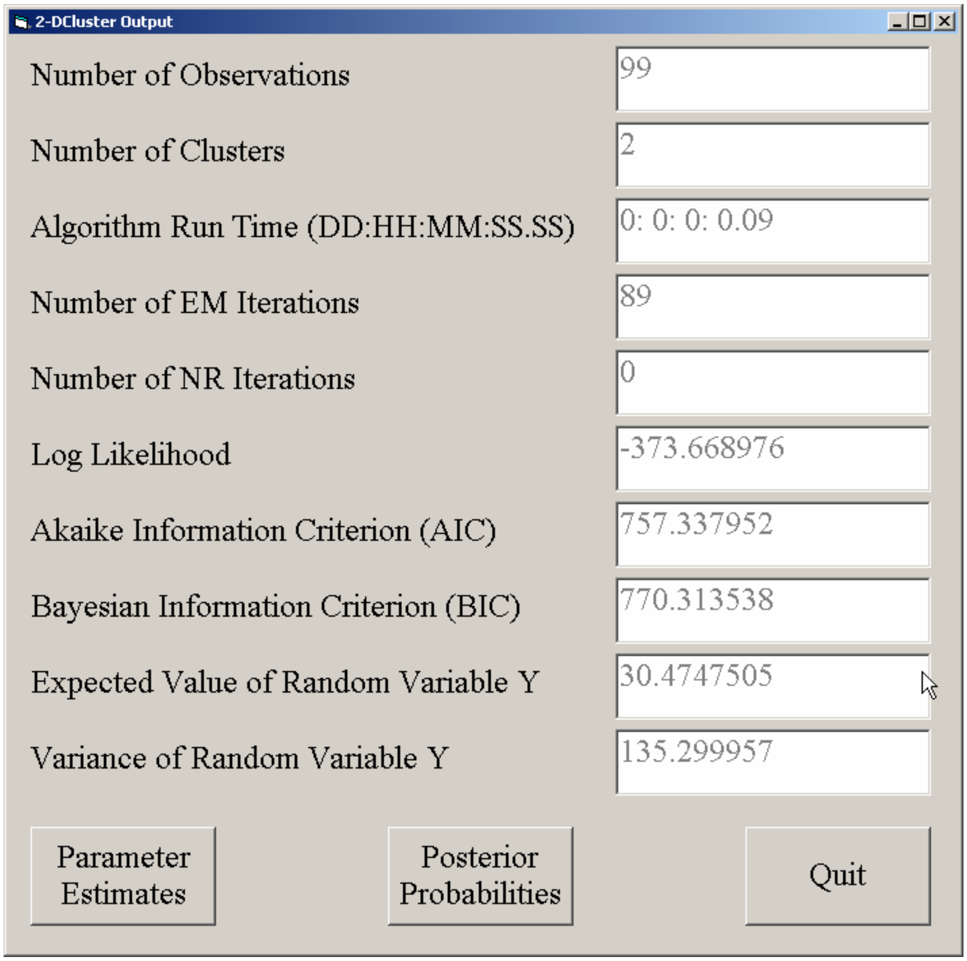
The three buttons on the bottom are activated by clicking on them. Run Algorithm starts the estimation process, View Documentation brings up a PDF version of this documentation file, and Quit stops the program. If run algorithm is selected, the following screen is displayed.



EM Algorithm Fortran Shell

Run times vary greatly. Everything should be ok as long as the hour glass is still present. Please be patient!
Click OK to Run the Algorithm!

This prompt indicates that the algorithm may take a while to converge. Click OK to start the algorithm. Once the algorithm converges, the following screen is displayed. If "bad" starting values were chosen, a screen indicating a numerical underflow or overflow may be seen. If this is the case, choose new starting values and/or algorithm control parameters and try again. In general, the starting values generated using a k-means cluster analysis work. However, large numbers of clusters, random starting values may work better.



2-DCluster Output

Number of Observations: 99

Number of Clusters: 2

Algorithm Run Time (DD:HH:MM:SS.SS): 0: 0: 0: 0.09

Number of EM Iterations: 89

Number of NR Iterations: 0

Log Likelihood: -373.668976

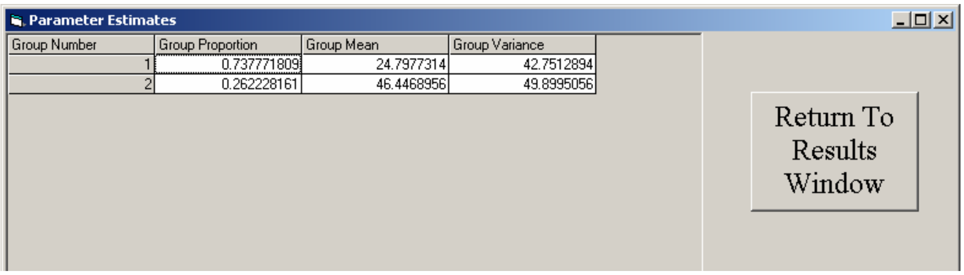
Akaike Information Criterion (AIC): 757.337952

Bayesian Information Criterion (BIC): 770.313538

Expected Value of Random Variable Y: 30.4747505

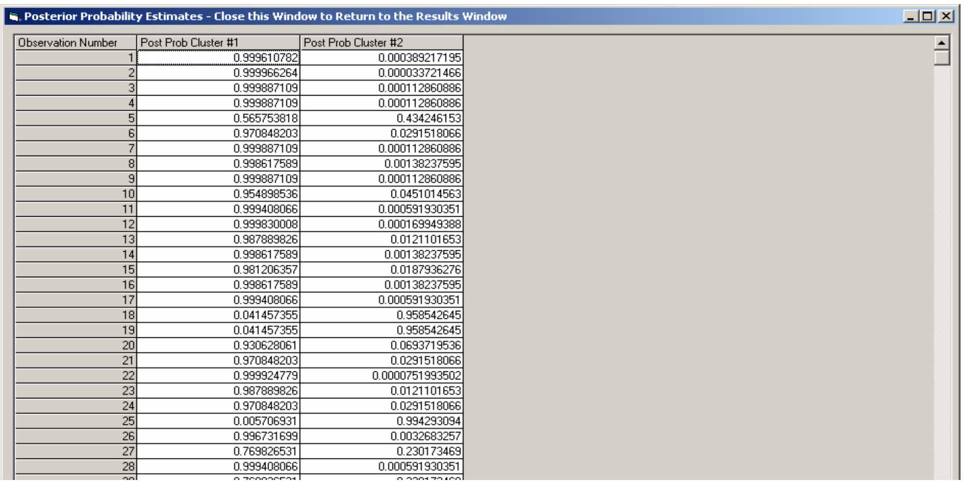
Variance of Random Variable Y: 135.299957

The output window indicates the number of observations and the number of clusters requested. The days, hours, minutes, seconds, and fractions of seconds are given in the algorithm run time output. The number of EM and NR iterations to obtain convergence are indicated. The number of NR iterations will be 0 unless the hybrid algorithm was selected. The log likelihood, Akaike information criterion (AIC), and Bayesian information criterion (BIC) for this model are indicated. The expected value and variance is given for the observations (represented by the random variable y). Clicking on parameter estimates gives the cluster specific parameter estimates. An example of this is shown below.



Group Number	Group Proportion	Group Mean	Group Variance
1	0.737771809	24.7977314	42.7512894
2	0.262228161	46.4468956	49.8995056

Clicking on posterior probabilities gives the posterior probability of each observation belonging to each cluster. An example of this is given below.


Posterior Probability Estimates - Close this Window to Return to the Results Window

Observation Number	Post Prob Cluster #1	Post Prob Cluster #2
1	0.999610782	0.000389217195
2	0.999966264	0.000033721466
3	0.999887109	0.000112860886
4	0.999887109	0.000112860886
5	0.565753818	0.434246153
6	0.970848203	0.0291518066
7	0.999887109	0.000112860886
8	0.998617589	0.00138237595
9	0.999887109	0.000112860886
10	0.954898536	0.0451014563
11	0.999408066	0.000591930351
12	0.999830008	0.000169949388
13	0.987889826	0.0121101653
14	0.998617589	0.00138237595
15	0.981206357	0.0187936276
16	0.998617589	0.00138237595
17	0.999408066	0.000591930351
18	0.041457355	0.958542645
19	0.041457355	0.958542645
20	0.930628061	0.0693719536
21	0.970848203	0.0291518066
22	0.999924779	0.0000751993502
23	0.987889826	0.0121101653
24	0.970848203	0.0291518066
25	0.005706931	0.994293094
26	0.996731699	0.0032683257
27	0.769826531	0.230173469
28	0.999408066	0.000591930351
29	0.769826531	0.230173469

For both the parameter estimates and the posterior probability estimates, left clicking anywhere in the grid exports the estimates to a Microsoft Excel worksheet. Quit terminates 2-DCluster. All of the estimates, including confidence intervals if requested, are output to files described in Section 2-d.5.

2-d.5 Data File Formats

All files referred to in this section are flat ASCII formatted files.

Data File Format (User Generated)

This file contains the observations to be clustered. The default name for this file is DATA.DAT. For unidimensional data, each line contains one observation. For two dimensional data, each line contains the row number of the observation, the column number of the observation, and the observed value separated by spaces.

Starting Value File Format (User Generated)

This file contains the starting values for the algorithm (if k-means or random starting values are not requested). Starting values for the proportions of observations falling in a given cluster (π), the cluster specific means (μ), and the cluster specific variances (σ^2) should be specified. This file contains one line for each cluster and is organized as:

$$\begin{array}{ccc} \pi_1 & \mu_1 & \sigma_1^2 \\ & \vdots & \\ \pi_C & \mu_C & \sigma_C^2 \end{array}$$

The number of clusters is indicated by C. The C proportions should sum to 1. The starting values for a given cluster are separated by spaces. The default name for this file is STARTVALS.DAT.

Parameter Estimate File Format (2-DCluster Generated)

The cluster specific parameter estimates are given in a file called *projbasename*.est. The default name for this file is PROJECT.EST. The proportions of observations falling in a given cluster (π), the cluster specific means (μ), and the cluster specific variances (σ^2) are reported. This file contains one line for each cluster and is organized as:

$$\begin{array}{ccc} \pi_1 & \mu_1 & \sigma_1^2 \\ & \vdots & \\ \pi_C & \mu_C & \sigma_C^2 \end{array}$$

The number of clusters is indicated by C. The C proportions should sum to 1. The estimates for a given cluster are separated by spaces.

Posterior Probability File Format (2-DCluster Generated)

The posterior probability estimates are given in a file called *projbasename*.pp. The default name for this file is PROJECT.PP. This file contains one number per line. The first C lines give the posterior probabilities for observation 1, the second C lines give the posterior probabilities for observation 2, etc. The number of clusters is indicated by C. The C posterior probabilities for a given observation should sum to 1.

Confidence Intervals for Parameter Estimates File Format (2-DCluster Generated)

This file is only generated if confidence intervals are requested. The filename is *projbasename*parmcis.est. This file contains one line for each cluster and is organized as:

$$\begin{array}{c}\pi_1 \;\; \pi_1^L \;\; \pi_1^U \;\; \mu_1 \;\; \mu_1^L \;\; \mu_1^U \;\; \sigma_1^2 \;\; \sigma_1^{2(L)} \;\; \sigma_1^{2(U)} \\ \vdots \\ \pi_C \;\; \pi_C^L \;\; \pi_C^U \;\; \mu_C \;\; \mu_C^L \;\; \mu_C^U \;\; \sigma_C^2 \;\; \sigma_C^{2(L)} \;\; \sigma_C^{2(U)}\end{array}$$

where $\pi_\#$ indicates the estimated proportion of observations falling in the cluster, $\pi_\#^L$ and $\pi_\#^U$ indicate the lower and upper confidence limits, $\mu_\#$ indicates the estimated cluster specific mean with confidence limits specified by $\mu_\#^L$ and $\mu_\#^U$, and $\sigma_\#^2$ indicates the estimated cluster specific variance with confidence limits specified by $\sigma_\#^{2(L)}$ and $\sigma_\#^{2(U)}$. The # indicates the cluster number (which is the same as the line number). All of the numbers are separated by spaces. The number of clusters is indicated by C. The C proportions should sum to 1.

Confidence Intervals for Posterior Probability Estimates File Format (2-DCluster Generated)

This file is only generated if confidence intervals are requested. The filename is *projbasename*ppcis.est. This file is organized as:

$$\begin{array}{c}\alpha_1^1 \;\; \alpha_1^{1L} \;\; \alpha_1^{1U} \\ \vdots \\ \alpha_1^C \;\; \alpha_1^{CL} \;\; \alpha_1^{CU} \\ \alpha_2^1 \;\; \alpha_2^{1L} \;\; \alpha_2^{1U} \\ \vdots\end{array}$$

$$\begin{array}{ccc} \alpha_2^C & \alpha_2^{CL} & \alpha_2^{CU} \\ & \vdots & \\ & \vdots & \\ \alpha_C^C & \alpha_C^{CL} & \alpha_C^{CU} \end{array}$$

where α_1^1 refers to the estimated posterior probability of observation 1 belonging to cluster 1 with α_1^{1L} and α_1^{1U} being the upper and lower confidence limits, respectively. The first C rows are the estimated posterior probabilities and confidence intervals for the first observation, the second C rows represent the same information for the second observation, etc. The numbers are separated by spaces.

Starting Values (2-DCluster Generated)

This file is only generated if k-means starting values are requested. The file contains the k-means starting values generated by 2-DCluster. The file is called *projbasename*.sv and contains one line for each cluster organized as:

$$\begin{array}{ccc} \pi_1 & \mu_1 & \sigma_1^2 \\ & \vdots & \\ \pi_C & \mu_C & \sigma_C^2 \end{array}$$

where π proportion indicates the proportion of observations falling in the clusters, μ indicates the cluster specific mean, and σ^2 indicates the cluster specific variance. The number of clusters is indicated by C. The C proportions should sum to 1. The staring values for a given cluster are separated by spaces.

Cluster Assignments (2-DCluster Generated)

This file indicates cluster assignments based on the maximum posterior probability for a given observation. The file is called *projbasename*.ca. Each line of the file contains the cluster number that the observation number (indicated by the row number of the file) is assigned. The cluster numbering is arbitrary.

A second file called *projbasename*.nc is also generated. This file contains one line for each of the C clusters. Two numbers separated by spaces appear on each line. The first number is the number of observations that were assigned to this cluster based on the maximum posterior probability. The second number is the proportion of observations assigned to this cluster. The proportions of observations assigned to each cluster should closely match the estimated proportions for the cluster specific parameters.

Other Files (2-DCluster Generated)

Two files are generated by 2-DCluster for internal use. These filenames are 2dcluster.prm and *projbasename*.s. These files may be ignored by the user.

2-d.6 Examples

Schizophrenia One Dimensional Normal Mixture

The first example is a one dimensional two component normal mixture. This data comes from a schizophrenia study reported by Levine (1981). He collated the results of seven studies on the age of onset of schizophrenia. Data was collected on 99 females and 152 males. (The analysis of the female data was used for the earlier screen shots.) The idea behind fitting a two component normal mixture model is that there are two general groups of schizophrenics.

The first group is diagnosed at a younger age and generally suffer from a more severe form of the illness. The second group is diagnosed later in life and generally has a less severe form of the illness. This data and a complete analysis may be found in Everitt *et al.* (2001). The 2-DCluster package includes the data and starting values used by Everitt.

The data for females is analyzed first. This data is found in women.dat and the starting values are in womenstartvals.dat. Analyzing the data using a 2 x 1 normal mixture model in 2-DCluster yields the results shown in the table below. The results reported by Everitt *et al.* (2001) are given for comparison's sake.

Schizophrenia Data for 99 Women

Parameter	Initial Value	Everitt Estimate	2-DCluster Estimate	2-DCluster 95% CIs
Proportion	0.5	0.74	0.74	(0.58, 0.89)
Mean 1	25	24.80	24.80	(22.42, 27.18)
Variance 1	10	42.75	42.75	(41.10, 44.40)
Mean 2	50	46.45	46.45	(40.52, 52.38)
Variance 2	10	49.90	49.90	(45.43, 54.37)

Notice that 2-DCluster reproduced Everitt's results. The EM algorithm took 89 iterations to converged and ran for less than a second. The log likelihood value was -373.67, the AIC value was 757.34, and the BIC value was 770.31. The expected value for the observations was 30.47 and the variance was 135.30. The two groups have quite different mean ages. Running the hybrid algorithm yielded the same results with 14 EM iterations and 6 NR iterations. The posterior probabilities and confidence intervals were also calculated but are not shown here.

The data for males is analyzed below. The data is found in men.dat and the starting values are in menstartvals.dat. Fitting the 2 x 1 normal mixture model in 2-DCluster yields the results shown in the table below.

Schizophrenia Data for 125 Men

Parameter	Initial Value	Everitt Estimate	2-DCluster Estimate	2-DCluster 95% CIs
Proportion	0.5	0.51	0.51	(0.34, 0.67)
Mean 1	25	20.25	20.37	(19.36, 21.38)
Variance 1	10	9.42	8.68	(7.68, 9.68)
Mean 2	50	27.76	27.61	(24.24, 30.98)
Variance 2	10	112.24	114.74	(112.73, 116.76)

Notice that the 2-DCluster results closely match those reported by Everitt. The EM algorithm took 131 iterations to converged and ran for less than a second. The log likelihood value was -428.10, the AIC value was 866.21, and the BIC value was 880.35. The expected value for the observations was 23.94 and the variance was 74.12. The two groups have quite different mean ages. The hybrid algorithm does not converge using these starting values. Notice that the average age of onset is earlier for the men than for the women. This concurs with the results reported in the schizophrenia literature. The posterior probabilities and confidence intervals were also calculated but are not shown here.

Simulated Two Dimensional Normal Mixture Distribution

A five component two dimensional normal mixture is simulated and analyzed below. The starting values are generated using a k-means cluster analysis. 1000 observations are simulated using the parameters given below. The data is contained in the file twod.dat.

Cluster #	Actual Parameters π	μ	σ^2	Parameter Estimates and 95% Confidence Intervals π	μ	σ^2
1	0.40	10.00	1.00	0.41 (0.38, 0.44)	10.05 (9.96, 10.15)	0.93 (0.87, 1.00)
2	0.20	20.00	2.00	0.21 (0.16, 0.26)	20.18 (20.00, 20.37)	1.89 (1.77, 2.01)
3	0.15	30.00	3.00	0.14 (0.12, 0.16)	30.07 (29.76, 30.38)	3.34 (3.12, 3.56)
4	0.10	40.00	4.00	0.09 (0.08, 0.11)	40.51 (40.10, 40.92)	3.50 (3.17, 3.82)
5	0.15	50.00	5.00	0.14 (0.12, 0.16)	50.16 (49.81, 50.50)	4.10 (3.81, 4.40)

The EM algorithm took 11 iterations to converged and ran for less than a second. The log likelihood value was -3179.81, the AIC value was 6387.63, and the BIC value was 6456.34. The expected value for the observations was 23.56 and the variance was 212.19. The estimates were very close to the actual values, and almost all of the 95% confidence intervals for the estimates contain the true values. Running the hybrid algorithm yielded almost identical results in 6 EM iterations and 3 NR iterations. The posterior probabilities and confidence intervals were also calculated but are not shown here.

2-d.7 Usage Notes

The algorithms are iterative in nature and the convergence time depends on many issues such as the sample size, the number of clusters, the tolerance value, the starting values, etc. The hybrid algorithm reduces the number of EM iterations required. However, for many problems, a single NR iteration takes as long to run as several EM iterations. For univariate problems, the hybrid algorithm generally converges more slowly than the EM algorithm. We recommend using the EM algorithm for these types of applications.

Appropriate starting values can be difficult to find. 2-DCluster provides starting values based on the results of a k-means cluster analysis. The EM algorithm often converges using these starting values. 2-DCluster also supports random starting values. Random starting values tend to result in convergence problems more frequently than the k-means starting values.

If there is no reason to suspect that the variances are different for the clusters, be sure to use the homogenous variance option. This option forces all the variance estimates to be the same and can result in accelerated convergence times.

Calculating confidence intervals requires the inversion of a 3C-1dimensional matrix, where C is the number of clusters. Models having substantial numbers of clusters require large matrices to be inverted, which can be very time consuming. For such applications, we recommend that the confidence intervals not be calculated.

2-DCluster was developed on a display with 1280 x 1024 resolution. Machines with lower resolution may have difficulty displaying the 2-DCluster graphical interface. If this occurs, users should try to increase their screen resolution as much as possible.

The error trapping in 2-DCluster is rudimentary. If you experience run time errors, please verify that the data files are in the format specified in Section 2-d.5. For some starting values and data sets, numerical underflows or underflows may occur. Most of these should be trapped. However, due to the iterative nature of the algorithm, some may not be trapped. In this situation, choosing new starting values or a smaller tolerance value generally allows the model to converge. Better error messages will be included in a future version of 2-DCluster.

Due to numerical precision issues, confidence intervals may not be calculated correctly for the proportions and posterior probabilities for models having large numbers of clusters. If confidence intervals are required for such a model, bootstrapping techniques could be employed. Another option is to recompile the fortran code using the quadruple precision option. This option doubles the precision of all operations involving real numbers, but sacrifices speed and increases memory requirements to do so.

2-DCluster includes the ability to run the same model multiple times using different data for input. In order to use this option, run 2-DCluster for the first model in the usual manner. Edit the file 2dcluster.prm in the 2-DCluster installation directory using any ASCII text editor. The second to last line should be 1. Change this number to the desired number of runs of the model and save the file. The data format is the same as that described in Section 2-d.5, with the data from the first model coming first, the data for the second model coming directly underneath, etc. In a sense, the data for multiple models is "stacked". Instead of running 2-DCluster through the usual graphical interface, run the program em.exe. Once the program terminates, the directory will contain the files described in Section 2-d.5 with a # appended to the file extension. For example, the file containing parameter estimates for model one could be project.est1, the estimates for model two project.est2, etc.

2-d.8 Author Contact Information

The author wishes to thank Dr. Viswanathan Ramakrishnan for his help in developing 2-DCluster. The author may be contacted through the Department of Biostatistics at Virginia Commonwealth University or by email at eharvey@mail2.vcu.edu. Dr. Ramakrishnan may be reached at vramesh@mail2.vcu.edu. Good luck with your research and please let us know if you have ideas for improving 2-DCluster.

2-d.9 References

Aitkin, M. & Aitkin, I. (1996). A Hybrid EM/Gauss-Newton Algorithm for Maximum Likelihood in Mixture Distributions. Statistics and Computing, 6, 127-130.

Everitt, B. S., Landau, S., & Leese, M. (2001). Cluster Analysis. (4th ed.) New York, NY: Oxford University Press, Inc.

Levine, R. (1981). Sex Differences in Schizophrenia: Timing or Subtypes? Psychological Bulletin, 90, 432-444.

Vita

Eric S. Harvey was born on June 20, 1974 in Richmond, Virginia and is an American citizen. He graduated from George Washington High School in Danville, Virginia in 1992. He attended Radford University in Radford, Virginia where he received the Dean’s Scholar Award and graduated in 1999 with a Bachelor of Science degree in Mathematics and Statistics. He later attended Strayer University in Midlothian, Virginia where he obtained his Master of Science degree in Information Systems in 2001. He and his wife Paige live in Midlothian, Virginia.