ABSTRACT

DASAH, JULIUS BERRY. Estimating the Number of Clusters in Cluster Analysis. (Under the direction of Dr. Dennis Boos and Dr. Leonard Stefanski.)

In many applied fields of study such as medicine, psychology, ecology, taxonomy and finance one has to deal with massive amounts of noisy but structured data. A question that often arises in this context is whether or not the observations in these data fall into some “natural” groups, and if so, how many groups? This dissertation proposes a new quantity, called the maximal jump function, for assessing the number of groups in a data set. The estimated maximal jump function measures the excess transformed distortion attainable by fitting an extra cluster to a data set. By distortion, we mean the average distance between each observation and its nearest cluster center. Distortion $d_g$ in the above sense, is a measure of the error incurred by fitting $g$ clusters to a data set. Three stopping rules based on the maximal jump function are proposed for determining the number of groups in a data set. A new procedure for clustering data sets with a common covariance structure is also introduced. The proposed methods are tested on a wide variety of real data including DNA microarray data sets as well as on high-dimensional simulated data possessing numerous “noisy” features/dimensions. Also, to show the effectiveness of the proposed methods, comparisons are made to some well known clustering methods.
Estimating the Number of Clusters in Cluster Analysis

by

Julius Berry Dasah

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

STATISTICS

Raleigh

2006

APPROVED BY:

Dennis Boos
Chair of Advisory Committee

Leonard Stefanski
Co-Chair of Advisory Committee

David Dickey

Jason Osborne
This work is dedicated to

*Judith and my parents*
Biography

Julius grew up in Wa, Upper West Region, Ghana. He later moved to Accra, where he received his Bachelor of Science degree in Mathematics and Statistics at the University of Ghana, Legon. He came to the United States in 2000, where he obtained his MS degree in Biostatistics from the University of Vermont in 2002. He joined the Department of Statistics, North Carolina State University in August, 2002 to pursue a Ph.D. degree in Statistics.
Acknowledgements

First, I am extremely grateful for having Dr. Dennis Boos and Dr. Len Stefanski as my advisors. Their knowledgeable guidance have significantly improved the contents of this thesis. I also admire their tireless energy and the devotion to their professional career. They are not only outstanding statisticians, but also great individuals.

My appreciation goes to my other committee members as well, Dr. Dave Dickey and Dr. Jason Osborne for their valuable comments, suggestions and encouragement. I thank Dr. Harry Daniels for agreeing to be the graduate school representative on my dissertation committee.

I thank Dr. Kim Weems, Dr. Cavell Brownie, Dr. Jacqueline Hughes-Oliver, Dr. Bill Swallow and Dr. Sastry Pantula for their support and encouragement over the past 4 years.

I also thank Adrian Blue and Terry Byron for their constant help and encouragement.

Finally, I would like to express my deepest gratitude and biggest appreciation to my parents, my sisters, Vivian, Sheila and Bernice and my dear wife, Judith for their unceasing support and love.
Contents

List of Tables vii
List of Figures ix

1 Introduction 1
  1.1 Background ........................................... 3

2 Determining the Number of Clusters 12
  2.1 Notation and Basic Definitions .......................... 12
  2.2 The Problem ............................................. 18
  2.3 The Jump Method ......................................... 25
    2.3.1 Motivation of the Jump Method: Rate Distortion Theory . . . . . 25
    2.3.2 Some Comments on the Method of Sugar and James (2003) .... 28
  2.4 The Maximal Jump Approach .............................. 32
    2.4.1 Computational Strategy: k-means .................... 38
  2.5 Model Based Clustering ................................ 41

3 Simulation Study 46
  3.1 Simulation 1: When No Noise Features are Present ............. 48
    3.1.1 Simulation 1 Design .................................. 48
    3.1.2 Simulation 1 Results ................................ 52
  3.2 Simulation 2: In the Presence of Noise Features ............... 54
    3.2.1 Simulation 2 Design .................................. 54
    3.2.2 Simulation 2 Results ................................ 56
  3.3 Simulation 3: Controlling Noise Features ...................... 59
    3.3.1 Simulation 3 Design .................................. 59
    3.3.2 Simulation 3 Results ................................ 59
  3.4 Simulation 4: Based on Covariance Structure ................... 60
    3.4.1 Simulation 4 Design .................................. 60
    3.4.2 Simulation 4 Results ................................ 61

4 Further Evaluations 63
  4.1 Cluster Validation ...................................... 63
  4.2 Comparing Two Partitions ................................. 64
  4.3 A Comparative Study ..................................... 69
    4.3.1 Evaluation Methodology ............................... 69
    4.3.2 Evaluation Metrics .................................. 73
    4.3.3 Artificial Data Set .................................. 74
    4.3.4 Factors Affecting the Performance of Method $MJD(\hat{c})$ .... 76
4.3.5 Comparing Method \( MJD(\hat{c}) \) to Method \( FR \) ........................................ 80

5 Clustering Data with a Common Covariance Structure ................................. 84
  5.0.6 Simulation ................................................................. 86
  5.0.7 Simulation Design ...................................................... 87
  5.0.8 Simulation Results ................................................... 88

6 Applications to Large \( p \), Small \( n \) Data Sets ........................................... 94
  6.1 Simulations ................................................................. 96
  6.2 Clustering Microarray Data Via The Maximal Jump Difference \( MJD(\hat{c}) \) Approach ................................................................. 99
    6.2.1 Data Sets ............................................................. 101
    6.2.2 Data Analysis ........................................................ 102

7 Applications ................................................................. 107
  7.1 Iris Dataset ............................................................... 107
  7.2 Ruspini Data ............................................................ 115

8 Conclusion and Discussions ........................................................................... 123

Bibliography ......................................................................................................... 127

A Proof of Proposition 2.4.1 ............................................................................ 136

B Derivation of Equation (2.17) ........................................................................ 142
List of Tables

2.1 Notation Used .......................................................... 17
2.2 Parameterizations of the covariance matrix $\Sigma_j$ and their geometric interpretations. .................................................. 43
3.1 Frequency of Prediction Error For the Five methods: $M_4^* \sim \mathcal{N}(0, a^2)$, $a = 1, 2, 3$, $p \sim DU[[2, \ldots, 10]]$, $p_I \sim DU[[1, \ldots, p]]$ .................................................. 53
3.2 Frequency of Prediction Error For the Five methods: Small Cluster Center Separations $M_1^*$ and $p_I \sim DU[[1, \ldots, p]]$ .................................................. 56
3.3 Frequency of Prediction Error For the Five methods: Medium Cluster Center Separations $M_2^*$ and $p_I \sim DU[[1, \ldots, p]]$ .................................................. 57
3.4 Frequency of Prediction Error For the Five methods: Large Cluster Center Separations $M_3^*$ and $p_I \sim DU[[1, \ldots, p]]$ .................................................. 58
3.5 Frequency of Prediction Error For the Five methods: Small Cluster Center Separation $M_{1}^*$ and $p \sim DU[[2, \ldots, 100]]$ .................................................. 60
3.6 Frequency of Prediction Error For the Five methods: Small Cluster Center Separation $M_1^*$ and $p \sim DU[[2, \ldots, 100]]$ and $p_I \sim DU[[1, \ldots, p]]$ .................................................. 62
4.1 The four possible patterns resulting from matching elements of two binary valued vectors .................................................. 65
4.2 Contingency table for comparing two partitions. .................................................. 66
4.3 Summary of notation used .................................................. 74
4.4 Description of variables in the Normalset data .................................................. 75
4.5 ANOVA for Normalset data .................................................. 79
4.6 Univariate statistics .................................................. 81
4.7 ANOVA for Normalset data: $MJD(\bar{c})$ versus $FR$ .................................................. 81
5.1 Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0$ in $\mathbf{R}$ and $\sigma_i = 1$, $(i = 1, \ldots, p)$ in $\mathbf{V}$ .................................................. 90
5.2 Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0.8$ in $\mathbf{R}$ and $\sigma_i = 1$, $(i = 1, \ldots, p)$ in $\mathbf{V}$ .................................................. 91
5.3 Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0$ in $\mathbf{R}$ and $\sigma_i = \sqrt{\frac{2i}{p+1}}$, $(i = 1, \ldots, p)$ in $\mathbf{V}$ .................................................. 92
5.4 Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0.8$ in $\mathbf{R}$ and $\sigma_i = \sqrt{\frac{2i}{p+1}}$, $(i = 1, \ldots, p)$ in $\mathbf{V}$ .................................................. 93
6.1 Simulation results: Large $p$, Small $n$ Situations .................................................. 99
7.1 Estimates of “Measures of Agreement” for Iris Data . . . . . . . . . . . 114
7.2 Estimated Number of Clusters for Ruspini Data Sets . . . . . . . . . . . 116
## List of Figures

2.1 Simulated data and estimated distortion from Example 2.2.1: (a) Simulated data; (b) Estimated distortion, \( \hat{d}_g \) ........................................... 23

2.2 The estimated Jump curve \( \hat{J}_g(p/2) \) for the simulated data set in Figure 2.1(a) .......................................................... 24

2.3 Simulated data, estimated distortion and the estimated Jump curve \( \hat{J}(p/2) \) curve from Example 2.3.2: (a) Simulated data containing \( g_0 = 3 \) elongated clusters; (b) Estimated distortion function \( \hat{d}_g \); (c) Estimated jump function \( \hat{J}_g(p/2) \) chose \( \hat{g} = 6 \) ............................................. 30

2.4 (a) Plot of \( \hat{H}(g, \hat{c}) \) versus Cluster number \( g \) for Example 2.2.1 (b) Plot of \( \hat{H}(g, \hat{c}) \) versus Cluster number \( g \) for Example 2.3.2 (c) Plot of \( \hat{W}(g, \hat{c}) \) versus Cluster number \( g \) for Example 2.2.1 (d) Plot of \( \hat{W}(g, \hat{c}) \) versus Cluster number, \( g \) for Example 2.3.2 (e) Plot of \( \hat{U}(g) \) versus Cluster number \( g \) for Example 2.2.1 (f) Plot of \( \hat{U}(g) \) versus Cluster number \( g \) for Example 2.3.2 ..................................................... 37

4.1 Performance of the \( MJD(\hat{c}) \) method: The smooth curves are lowess fits (with \( f=0.5 \)). Top panel: left:-Adjusted Rand index versus Actual number of clusters. center:-Fowlkes-Mallows index versus Actual number of clusters. right:-AEPCN versus Actual number of clusters. Middle panel: left:-Adjusted Rand index versus Dimension. center:-Fowlkes-Mallows index versus Dimension. right:-AEPCN versus Dimension. Bottom panel: left:-Adjusted Rand index versus Important dimensions. center:-Fowlkes-Mallows index versus Important dimensions. right:-AEPCN versus Important dimensions ...................................................... 77

4.2 Performance of the \( MJD(\hat{c}) \) method: The smooth curves are lowess fits (with \( f=0.5 \)) Top panel: left:-Adjusted Rand index versus \( DetRho \). center:-Fowlkes-Mallows index versus \( DetRho \). right:-AEPCN versus \( DetRho \). Middle panel: left:-Adjusted Rand index versus \( SdSigma \). center:-Fowlkes-Mallows index versus \( SdSigma \). right:-AEPCN versus \( SdSigma \). Bottom panel: left:-Adjusted Rand index versus \( SEPINDEX \). center:-Fowlkes-Mallows index versus \( SEPINDEX \). right:-AEPCN versus \( SEPINDEX \) ..................................................... 78
4.3 $MJD(\hat{c})$ versus $FR$: The smooth curves are lowess fits (with $f=0.5$) Top panel: left:-Mean difference in Adjusted Rand index versus Dimension center:-Mean difference in Fowlkes-Mallows index versus Dimension right:-Mean difference in $AEPCN$ versus Dimension Middle panel: left:-Mean difference in Adjusted Rand index versus Important dimensions center:-Mean difference in Fowlkes-Mallows index versus Important dimensions right:-Mean difference in $SEPINDEX$ versus Important dimensions Bottom panel: left:-Difference in Adjusted Rand index versus $SEPINDEX$ center:-Difference in Fowlkes-Mallows index versus $SEPINDEX$ right:-Difference in $AEPCN$ versus $SEPINDEX$. 

5.1 Performance of $FR$ for cases in which $p = 0.8$ in $R$ and $\sigma_i = 1$, $i = 1, \ldots, p$: (a) Plot of mean adjusted Rand index versus dimension $p$; (b) Plot of mean $AEPCN$ versus dimension $p$. 

6.1 Clustering the yeast cell data set (Yeung et al., 2001): (a) Clustering genes via $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (b) Clustering samples via $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (c) Clustering the genes via $FR$, a plot of BIC versus number of clusters $g$. 

6.2 Clustering the colon data set via $MJD(\hat{c})$: (a) Clustering samples, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (b) Clustering genes, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$. 

6.3 Clustering the leukemia data set via $MJD(\hat{c})$: (a) Clustering samples, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (b) Clustering genes, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$. 

7.1 Iris data: Sepal length versus Sepal width. 

7.2 Pairs plot of the Iris data. 

7.3 Cluster number estimates of the Iris data via all three of our proposed methods: (a) $MJ(\hat{c})$, a plot of $\hat{H}(g, \hat{c})$ versus $g$; (b) $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus $g$; (c) $MJD(0)$, a plot of $\hat{U}(g)$ versus $g$. 

7.4 BIC versus cluster number. 

7.5 Estimate of the cluster number of the iris data without sepal length. 

7.6 Ruspini data: (a) Plot of the Ruspini data set; (b) Cluster number estimate via $MJ(\hat{c})$, a plot of $\hat{H}(g, \hat{c})$ versus cluster number $g$; (c) Cluster number estimate via $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus cluster number $g$; (d) Cluster number estimate via $MJD(0)$, a plot of $\hat{U}(g)$ versus cluster number $g$. 

7.7 Cluster number estimate of the modified Ruspini data with $p_n$ extra "noisy" features via the $SJ$ method: Plots of estimated jump $\hat{J}_g(p/2)$ versus cluster number $g$, with $p = 2 + p_n$ where, (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 

7.8 (a) Cluster number estimate of the modified Ruspini data with $p_n$ extra "noisy" features via the $FR$ method: Plots of BIC versus cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 

7.9 Cluster number estimate of the modified Ruspini data with $p_n$ extra "noisy" features via our proposed $MJ(\hat{c})$ method: Plots of $\hat{H}(g, \hat{c})$ versus cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 

x
7.10 Cluster number estimate of the modified Ruspini data with $p_n$ extra
“noisy” features via our proposed $MJD(\hat{c})$ method: Plots of $\hat{W}(g, \hat{c})$
versus cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$.

7.11 Cluster number estimate of the modified Ruspini data with $p_n$ extra
“noisy” features via our proposed $MJD(0)$ method: Plots of $\hat{U}(g)$ versus
cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 121 122
Chapter 1

Introduction

In many applied fields of study one has to deal with massive amounts of “noisy” but structured data. A question that often arises is whether or not the observations in these data fall into some “natural” groups, and if so, how many groups? This question arises in many disciplines including medicine, economics, psychology, ecology, and taxonomy. A number of techniques, including model-based approaches and various clustering algorithms, have been proposed to answer the above question.

The term “clustering” is typically used to describe the partitioning of a set of objects into groups or “clusters” so that objects within a group are more “identical” or “similar” to one another than objects in different groups or the grouping of items in a data set “by some natural criterion of similarity” (Murray and Estivill-Castro, 1998). It is evident from the above definition that similarity (or dissimilarity) is fundamental to the definition of a cluster.

Most clustering procedures try to measure the “similarity” between objects
and then proceed to group the objects so as to maximize within-class similarity. Unfortunately, the question of what constitutes “similarity” is a subject of great controversy. This controversy in part accounts for the existence of numerous clustering methods available today. Authors often define measures of “similarity” to reflect some specific philosophical view point. It is therefore not surprising that no clustering technique has been found to be universally applicable in uncovering the variety of structures that are present in multidimensional data sets.

Traditionally, clustering algorithms have been categorized into hierarchical and partitional clustering. Hierarchical clustering employs a technique of successively merging small clusters into larger clusters (Bottom-up, Agglomerative) or successively splitting larger clusters into smaller ones (Top-down, Divisive). Perhaps the most commonly used hierarchical clustering algorithm is hierarchical agglomerative clustering (HAC). HAC starts with one datum per cluster and then recursively merges two clusters with the smallest distance between them into a larger cluster until only one cluster is left. The algorithms tend not to be helpful in dealing with large data sets.

On the other hand, partitional methods tend to directly seek a partition that optimizes some predefined numerical (distance) measure. Unlike hierarchical clustering, partitional algorithms requires the specification of a predetermined cluster number by the user.

Recent years have seen a surge in the applications of cluster analysis in many fields, including statistics, pattern-analysis, machine intelligence, and more recently, knowledge recovery and data mining. The importance and interdisciplinary nature of
Cluster analysis is evident through its vast literature. The continued influx of new clustering techniques in the literature has made the task of choosing an appropriate clustering method a very difficult one. But perhaps the most serious issue that confronts a researcher in a real life situation is the choice of the right number of clusters, $g$, to include in the final solution.

Clustering algorithms are widely known to be useful tools for data mining, compression, probability density estimation, as well as for many other tasks. However, many of these clustering algorithms (partitional) require the user to specify the number of clusters, $g$, which typically is unknown. The problem of choosing the appropriate number of clusters $g$, as noted by many authors including, Ling (1971) and Sneath and Sokal (1973), is an important but difficult one. Everritt (1979) and also Hartigan (1985) have identified the problem of finding the right number clusters as an unresolved problem in cluster analysis that may not be capable of a full solution. Determining the right number of clusters in a data set often involves ad hoc decisions based on prior knowledge, assumptions and practical experience.

1.1 Background

In this Section, we briefly survey some of the more commonly used clustering methods for determining the number of clusters embedded in a data set. This survey is not meant to be a comprehensive review, but rather is aimed at grouping these methods along similar motivations. A more comprehensive survey of the topic may be found in books like Jain and Dubes (1998).
In past decades, the research into determining the “optimal” number of clusters, \( g \), in a data set have been along three major lines. The first is to formulate the cluster number selection problem as the choice of the number of components in a finite mixture model. The clustering problem then is to estimate the number of mixture components and the parameters of each component. In addition to the observed variables for a data item, \( \mathbf{X}_i (i = 1, \cdots, n) \), we assume there is an unobserved variable indicating “cluster membership” of the given data item. That is, the data are assumed to come from some mixture model, \( M \), where mixing labels are hidden or missing. The parameters of the model \( M \), say \( \theta(g) \), may be obtained via the so called classification maximum likelihood procedure (Basford and McLachlan, 1988; Banfield and Raftery, 1993).

Under the assumption that the data set is a mixture of multivariate normals, Symons and Scott (1971) derived the maximum likelihood (ML) estimate of the hidden mixing labels. By placing various constraints on the covariance structure of the mixture of multivariate normals, they showed that their maximum likelihood (ML) estimate was equivalent to several well known standard clustering methods, including the sum of squares criterion (Duda and Hart, 1973) and the criterion of Friedman and Rubin (1967).

The form of a finite mixture density with \( g \) subpopulations or clusters is

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

(1.1)

where the \( \pi_j \)'s are called the mixing proportions and represent the probabilities for each subpopulation. The \( \pi \)'s are all non-negative and sum to one. Clustering in the above
sense, is then an optimization problem involving the estimation of the parameters, \( \hat{\theta} \), associated with the above mixture density, where the probability density specified by each cluster could take any form. By virtue of their flexibility, finite mixtures provide an excellent modeling tool to describe situations in which each observation in a sample may be viewed as being generated from one of a finite set of alternative mechanisms. However, as noted by Mario et al. (1999), strict adherence to the above interpretation is not required. The Expectation-Maximization (EM) algorithm is a well known method for estimating the parameters of mixture models (Dempster et al., 1977). The usefulness of finite mixture models is not limited to clustering applications. Mixtures are also ideal for representing complex class conditional pdf’s in supervised learning.

From a purely computational view point, most methods for determining the number of components or groups in a mixture can be categorized as either deterministic or stochastic. Deterministic methods start by obtaining a set of candidate models for a range of values \( g \), say from \( g_{\text{min}} \) to \( g_{\text{max}} \), where the optimal \( g \) is known, or thought, to belong. Then the optimal \( g \) is obtained by

\[
\hat{g} = \arg\min_g \left\{ C \left( \hat{\theta}(g), g \right) \right\}, \quad g = g_{\text{min}}, \ldots, g_{\text{max}}
\]

where \( C \left( \hat{\theta}(g), g \right) \) is some model selection criteria and \( \hat{\theta}(g) \) is the estimate of the mixture model parameters, assuming \( g \) mixture components. Typically,

\[
C \left( \hat{\theta}(g), g \right) = - \log p \left( x | \hat{\theta}(g) \right) + \log P(g),
\]

where \( P(g) \) is an increasing function penalizing larger values of \( g \).
Consequently, model selection criteria such as the Akaike’s Information Criterion (AIC) (Scolve, 1983; Bozdogan and Scolves, 1984) and more recently the Bayesian Information Criterion (BIC) (Banfield and Raftery, 1993; Fraley and Raftery, 1998) have been proposed in the clustering literature for determining the number of clusters. It is, however, well known that these methods have the tendency to overestimate or underestimate the correct number of clusters due to the difficulty of selecting an appropriate penalty function. It is common practice to plot these model selection criteria against the number of clusters $g$, in identifying the “true” number of clusters.

Stochastic methods on the other hand rely heavily on Markov Chain Monte Carlo (MCMC) sampling. MCMC can either be used to implement model selection criteria directly to obtain the estimate of $g$ (Bensmail et al., 1997; Roeder and Wasserman, 1997), or in a fully Bayesian way by sampling from the full posterior distribution with $g$ considered unknown (Richardson and Green, 1997). Despite their formal appeal, MCMC based methods tend to be computationally demanding and not widely used in applications.

A number of authors, mainly in the 70’s and 80’s, attempted to derive formal tests of the number of clusters. The problem with most statistical tests for clusters is that they tend to have intractable sampling distributions or involve null hypotheses for which rejection is uninformative. A common null hypothesis in testing for clusters in a data set is that the data come from the multivariate normal distribution for each component. For instance, Wolf (1971) proposed a likelihood ratio criterion for testing the hypothesis of $g$ clusters against $g - 1$ clusters. This method is based on the assumption
of multivariate normality. The multivariate normal null hypothesis arises naturally in normal mixture models (Titterington, Smith, and Makov, 1985; McLachlan and Basford, 1988). Unfortunately, the likelihood ratio test statistic does not have the usual asymptotic chi-squared distribution because the regularity conditions do not hold. Wolf (1978) suggests approximations to the asymptotic distribution of the likelihood ratio test statistic. The value of these approximations is, however, not clear (Everitt, 1981; Thode, Mendell, and Finch, 1988). For small samples, McLachlan and Basford (1988) propose bootstrapping the likelihood ratio test. The multivariate normal null hypothesis tends to have a high probability of rejection if the data are sampled from a distribution with lower kurtosis than a normal distribution, such as a uniform distribution (Engleman and Hartigan 1969).

Engleman and Hartigan (1969) derived a likelihood ratio test to test the hypothesis of $H_0 : g = 1$ versus $H_1 : g = 2$ for the univariate normal case with equal variances. This result was generalized by Hartigan (1978) who considered the alternative hypothesis $H_1 : g > 2$. Lee (1979), provides a multivariate extension to the univariate formulation of Engleman and Hartigan (1969) likelihood ratio test. Beale (1969) provides an ‘F test’, that may be used to test whether or not a data set divided into $g_1$ clusters is significantly better than a subdivision into $g_2$ ($g_1 < g_2$) clusters. However, as noted by Everitt (1979), this statistic is useful only when the clusters are fairly well separated and hyper-spherical.

Other formal tests of the number of clusters can be found in Day (1969), Scott and Symons (1971) and Hartigan (1975). These tests tend to be very sensitive to the
normality assumption concerning the group densities.

The use of the uniform distribution as an appropriate null hypothesis has also been suggested in the literature (Hartigan, 1978; Arnold, 1979; Sarle, 1983). Here the data are assumed to be sampled from the uniform distribution. This null hypothesis typically leads to conservative error rates when the data are sampled from a strongly unimodal distribution like the normal distribution. Results produced by the uniform null can be very sensitive to the shape of the region of support, especially with higher dimensional data sets. Some asymptotic results are available for the within-cluster sum of squares (Hartigan 1978, 1985; Pollard 1981; Bock 1985).

Nonparametric tests for the number of clusters based on nonparametric density estimates are plentiful (Silverman, 1986; Mueller and Sawitzki, 1991; Minnotte, 1992; Polonik, 1993). The only assumptions usually made here are that the observations are sampled independently and that the distribution can be estimated nonparametrically. Hartigan and Hartigan (1985) and Hartigan (1985) developed a test of unimodality versus bimodality in the univariate case. Here clusters correspond to modes in the probability density function.

The third line of methods estimates the “optimal” number of clusters in a data set based on procedures that are largely heuristic but intuitively reasonable. As echoed by Fraley and Raftery (1998), the third line tend to be based “on algorithms and criteria in the belief that intuitively reasonable criteria should produce good results over a wide range of possible (and generally unstated) models.” Most commercial statistical software tend to be based on this type. Below is an overview of some these heuristic
Everitt (1969) credits an early attempt at solving the problem of finding the right number of clusters in an arbitrary data set to Thorndike (1953). Thorndike plotted the average within-cluster distance against number of clusters, \( g \), and proceeded to determine the optimal number of clusters as the point where a marked flattening of the curve is observed. Similar procedures have been proposed by many authors, where the optimization criterion is plotted against the number of clusters. According to Gower (1975), “a sharp step in this plot indicates the number of classes; otherwise there is no justification of containing more than one class.” However, in practice, the decision as to whether such plots do contain “sharp steps” can be highly subjective and hence is not particularly helpful in many applications (Everitt, 1979).

Marriot (1971) adopts a less subjective approach. He suggests finding the number of clusters in a data set by taking the value of \( g \) that minimizes \( g^2 |W| \), where \( |W| \) is the determinant of the pooled within cluster sum of squares. In practical applications, one would hope that a unimodal distribution will result in a minimum value of \( g = 1 \) and for group distributions, the minimum will indicate the appropriate value of \( g \). Calinski and Harabasz (1974) determine the number of clusters, \( g \), by

\[
\hat{g}_C = \max_g \left\{ \frac{\text{trace}(B)}{g - 1} \bigg/ \frac{\text{trace}(W)}{n - g} \right\},
\]

where \( W \) and \( B \) are the sample within and between group covariance matrices respec-
tively. Hartigan (1975) suggests choosing the smallest value of $g$ such that

$$HN(g) = (n - g - 1) \left[ \frac{W(g)}{W(g + 1)} - 1 \right] \leq 10$$

as the “true” number of clusters. Here, $W(g)$ is the within-cluster sum of squares with $g$ clusters. Krazanowski and Lai (1985), estimated the number of clusters by maximizing

$$KL(g) = \left| \frac{DIFF(g)}{DIFF(g + 1)} \right|,$$

where $DIFF(g) = (g-1)^{p/2} W(g-1) - g^{p/2} W(g)$. This statistic is, in a sense a measure of the rate of change of distortion, taking into account the dimension of the space, $p$.

The silhouette statistic defined by

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]},$$

where $a(i)$ is the mean distance between the $i$th point and all other observations in its cluster, and $b(i)$ is the mean distance to points in the “nearest” cluster, was proposed by Kaufman and Rousseeuw (1990) as useful statistic for determining the “true” number of clusters in a data set. “Nearest” here is in the sense of the cluster minimizing $b(i)$. They suggest choosing the number of clusters, $g$, that maximizes the average values of $s(i)$. Large values of $s(i)$ tend to indicate strong clustering. Tibshirani et al. (2001) proposed the Gap statistic defined by

$$Gap(g) = \frac{1}{B} \sum_b \log (W^* s(b)) - \log W(g),$$

10
where \( \log(W^*_b(g)) \) is the within-cluster sum of squares for the \( b \)th uniform data set. The method essentially compares the \( \log W(g) \) curve to a similar curve derived from an average of \( B \) data sets uniformly distributed over a rectangle containing the data. This method automatically determines the optimal number of clusters by locating where the gap between the two curves is largest. The estimate of the “true” number of clusters is given by

\[
\hat{g} = \arg\min_g \left\{ g | \text{Gap}(g) \geq \text{Gap}(g + 1) - s'_{g+1} \right\},
\]

where \( s'_g = s_g \sqrt{1 + B^{-1}} \) and \( s_g \) is the standard deviation of \( \log W(g) \) over \( B \) simulations. The method is said to work well for the case where the optimal number of clusters is one. This is a case where most methods fail.

An information theoretic approach for finding the number of clusters in a data set has been proposed by Sugar and James (2003). We will discuss this approach in detail in the next chapter. The model-based clustering method of Fraley and Raftery (1988), which has received much attention in recent years will also be discussed in the next chapter.
Chapter 2

Determining the Number of Clusters

We present some common notation and basic definitions used throughout this dissertation in Section 2.1. We formulate the clustering problem of determining the number of “natural” groups, if any, present in a data set in the most general context in Section 2.2. We then proceed to discuss the jump method of Sugar and James (2003) for determining the number of clusters \( g \), in a data set in Section 2.3. We present our approach to the cluster number problem in Section 2.4, and finally we discuss the model-based clustering method of Fraley and Raftery (1998) in Section 2.5.

2.1 Notation and Basic Definitions

Let \( X_1, \ldots, X_n \) denote a random sample of size \( n \), where \( X_i \) is a \( p \)-dimensional random vector with distribution function \( F(x) \) on \( \mathbb{R}^p \). Now, let \( \mathcal{X} = (X_1^T, \ldots, X_n^T)^T \), where superscript \( T \) denotes vector transpose. Note that \( \mathcal{X} \) is an \( n \times p \) matrix containing the entire sample.
We assume the distribution function $F(x)$ of $X_i$ can be written in the form

$$F(x) = \sum_{j=1}^{g} \pi_j F_j(x)$$  \hspace{1cm} (2.1)

where

$$0 \leq \pi_j \leq 1, \quad (j = 1, \ldots, g)$$

and

$$\pi_1 + \cdots + \pi_g = 1.$$

Equation (2.1) is often referred to as a $g$-component finite mixture distribution. We associate with the $p$-dimensional random vector $X$ having the distribution function $F(x)$, a latent indicator variable $Z$ defined by

$$Z = \sum_{j=1}^{g} j I(X \text{ “drawn” from } F_j(x))$$  \hspace{1cm} (2.2)

where $I(.)$ is an indicator function, so that

$$Pr(X \leq x|Z = j) = F_j(x).$$  \hspace{1cm} (2.3)

Note that the $(p+1)$-dimensional random vector $(X^T, Z)^T$ may be regarded in EM algorithm language as a “complete data” point, where $X$ constitutes the observable $p$-dimensional random vector and $Z$ the unobservable indicator variable. Also note that, the latent indicator variable $Z$ defines the component in the mixture model (2.1) from which the random vector $X$ is supposed to have arisen.
In model-based clustering, the $g$ components are often modeled using the multivariate normal distribution provided the data are continuous data. That is, the density of the $j$th mixture component density is taken to be

$$f_j(X|\mu_j, \Omega_j) = \frac{\exp\left\{-\frac{1}{2}(X - \mu_j)^T\Omega_j^{-1}(X - \mu_j)\right\}}{(2\pi)^{\frac{p}{2}}|\Omega_j|^{\frac{1}{2}}}$$ (2.4)

where $\mu_j$ and $\Omega_j$ are respectively the mean vector and the covariance matrix $X$. The $j$th component is commonly referred to as the $j$th cluster in model-based clustering. It is also common to refer to $\mu_j$ and $\Omega_j$ respectively as the mean vector and covariance structure of the $j$th cluster. The number of mixture components $g$, in this context is the number of clusters.

We define the following commonly used terms:

- **“True” cluster number**: We define the “true” cluster number of the data set $X$ as

  $$g_0 = \sup\{\text{supp}(Z)\},$$

  where $\text{supp}(Z)$ denotes the support of $Z$.

- **The $j$th cluster**: For $i = 1, \ldots, n$, let $(X_i^T, Z_i)$ be iid copies of $(X^T, Z)$. By the $j$th cluster we mean the set

  $$\mathcal{X}_j = \{X_i : Z_i = j, i = 1, \ldots, n\}$$ (2.5)

Let $n_j$ be the number of items in the set $\mathcal{X}_j$, i.e., $n_j$ is the cardinality of $\mathcal{X}_j$. 
We shall associate with $X_j$ the following parameters of interest:

- A $p \times 1$ vector $c_j$, a measure of location of the $n_j$ items in $X_j$, commonly referred to as the center or centroid of $X_j$.

- A $p \times p$ matrix $\Sigma_j$, that determines other geometric features associated with the points in $X_j$. This $\Sigma_j$ is commonly referred to as the covariance structure of $X_j$.

Specific definitions of the above parameters tend to vary from method to method. For instance, in model-based clustering, the cluster center $c_j$ and the covariance structure $\Sigma_j$ of $X_j$ are often taken to be respectively the mean $\mu_j$ and the covariance matrix $\Omega_j$, of the $j$th component of a finite mixture. The k-means algorithm on the other hand, assumes a common covariance structure, the identity matrix (i.e $\Sigma_j = I_p \forall j$), and adopts the following definition for $c_j$:

$$c_j = E(X|Z_i = j).$$

(2.6)

Given that the $Z_i$’s are known, the k-means algorithm estimates (2.6) by

$$\hat{c}_j = \frac{\sum_{i=1}^{n_i} 1(Z_i = j)X_i}{\sum_{i=1}^{n_i} 1(Z_i = j)},$$

(2.7)

assuming the denominator is greater than 0.

In some applications of finite mixture models, the number of mixture components $g$, is assumed to be known. However in a clustering context, $g$ is often unknown and thus has to be inferred from the data available, along with other parameters as-
sociated with the mixture model. Throughout this manuscript, we will assume $g_0$ is the actual or “true” number of mixture components (cluster number). If the number of groups or components in a data set is known \textit{a priori}, we will refer to that number as the “true” cluster number $g_0$.

To study the performance of clustering methods, it is a common practice to use labeled data. Most authors use a sampling design known as the separate sampling design in obtaining labeled data (McLachlan and Peel, 2000). Under this design, for each group corresponding to the $j$th component ($j = 1, \ldots, g$), we separately draw $n_j$ items. The combined sample size is $n = \sum_{j=1}^{g} n_j$. In this context, for $j = 1, \ldots, g$, the $n_j$ observations drawn from group $j$ constitute a set $X_j$ (the $j$th cluster).

This separate sampling design is appropriate in retrospective studies that are common in epidemiology. We should, however, note that under this design, the observed proportions of the items from the mixture components do not correctly estimate the mixing proportions $\pi_j$’s, which of course is of a lesser concern in clustering.

Most of the notation used in this dissertation is in the context of separate sampling, as this is the most common sampling technique for generating data for the purpose of evaluating clustering methods. Notation used in this dissertation is summarized in Table 2.1.
Table 2.1: Notation Used

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>A $p$-variate random vector</td>
</tr>
<tr>
<td>$X_i$</td>
<td>Observed data. Here, $X_i$ is the $i$th observation</td>
</tr>
<tr>
<td>$g$</td>
<td>An arbitrary cluster number</td>
</tr>
<tr>
<td>$g_0$</td>
<td>“True” cluster number</td>
</tr>
<tr>
<td>${1,2,\ldots,g_{\text{max}}}$</td>
<td>Set of candidate cluster numbers</td>
</tr>
<tr>
<td>$n$</td>
<td>Sample size</td>
</tr>
<tr>
<td>$p$</td>
<td>Dimension or number of features of data</td>
</tr>
<tr>
<td>$p_I$</td>
<td>Number of important features or dimensions</td>
</tr>
<tr>
<td>$X_j$</td>
<td>The $j$th cluster</td>
</tr>
<tr>
<td>$c_j$</td>
<td>Cluster center of $X_j$, the $j$th cluster</td>
</tr>
<tr>
<td>$R_j$</td>
<td>A $p \times p$ AR1 correlation matrix with parameter $\rho_j$</td>
</tr>
<tr>
<td>$V_j$</td>
<td>A $p \times p$ diagonal matrix</td>
</tr>
<tr>
<td>$\Sigma_j = V_j R_j V_j$</td>
<td>Covariance structure of $X_j$</td>
</tr>
<tr>
<td>$n_j = \text{Card}(X_j)$</td>
<td>Number of points in $X_j$</td>
</tr>
<tr>
<td>$N_{g_0} = {(n_1,\ldots,n_{g_0}) : \sum_{j=1}^{g_0} n_j = n}$</td>
<td></td>
</tr>
<tr>
<td>$d_g$</td>
<td>Distortion associated with $g$ clusters</td>
</tr>
<tr>
<td>$J(g,t,c)$</td>
<td>Jump function</td>
</tr>
<tr>
<td>$H(g,c)$</td>
<td>Maximal jump function</td>
</tr>
<tr>
<td>$\tau_g$</td>
<td>$J(g,\tau_g,c) \geq J(g,t,c)$ $\forall t &gt; 0$ given a fixed scalar $c$</td>
</tr>
</tbody>
</table>

* Population quantities
2.2 The Problem

Let \( \mathcal{X} = (X_1^T, \ldots, X_n^T)^T \) be a sample of \( p \)-dimensional observations. Geometrically, this sample may be viewed as points in a \( p \)-dimensional space. Suppose our goal is to partition \( \mathcal{X} \) into \( g \) disjoint subsets \( \mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_g \), such that the data points within each subset, \( \mathcal{X}_j (j = 1, \ldots, g) \), representing a cluster, tend to be “closer” to each other than data points in different clusters. Then two immediate questions of relevance as raised by Marriott(1971) are:

1. What is the best subdivision of \( \mathcal{X} \) for a given number, \( g \) of groups?

2. What is the best value of \( g \)?

To answer the first question, one way to proceed is to define some objective function that measures the clustering quality of a given partition of the sample into \( g \) groups. The problem then reduces to one of finding the partition that optimizes the defined objective function.

Concepts of homogeneity and separation are often used to develop a good objective function. By homogeneity, we mean objects in a group have a cohesive structure, and by separation, we mean groups are well isolated from each other.

For the second question, the derived objective function is often successively evaluated for \( g = 1, 2, \ldots, g_{\text{max}} \) partitions (groups) and the “best” value of \( g \) selected. The “best” value of \( g \), is usually taken to be the value that optimizes the defined objective function. This process is often called the “stopping rule” for the associated
Though various objective functions for clustering have been proposed in the literature, a commonly used objective function is a measure of within cluster dispersion called the distortion function.

**Definition 2.2.1** Let \( \mathbf{X}, \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_g \in \mathbb{R}^p \), and suppose that \( \Gamma \) and \( \Sigma \) are \( p \times p \) positive definite matrices. Let \( \mathbf{M} = (\mathbf{m}_1^T, \mathbf{m}_2^T, \ldots, \mathbf{m}_g^T) \). For \( j = 1, \ldots, g \), define the indicator function \( \mathbb{I}_j \) by

\[
\mathbb{I}_j(\mathbf{X}) = \mathbb{I}\{(\mathbf{X} - \mathbf{m}_j)^T \Gamma^{-1} (\mathbf{X} - \mathbf{m}_j) = \min_{1 \leq s \leq g} [(\mathbf{X} - \mathbf{m}_s)^T \Gamma^{-1} (\mathbf{X} - \mathbf{m}_s)] \}
\]

and

\[
c(\mathbf{X}, \mathbf{M}, \Gamma) = \mathbf{m}_1 \mathbb{I}_1(\mathbf{X}) + \sum_{j=2}^g \left[ \mathbf{m}_j \mathbb{I}_j(\mathbf{X}) \prod_{k=1}^{j-1} (1 - \mathbb{I}_k(\mathbf{X})) \right].
\]

Also let

\[
r(\mathbf{X}, \mathbf{M}, \Gamma, \Sigma) = \{\mathbf{X} - c(\mathbf{X}, \mathbf{M}, \Gamma)\}^T \Sigma^{-1} \{\mathbf{X} - c(\mathbf{X}, \mathbf{M}, \Gamma)\}.
\]

Suppose \( \mathbf{X} \) is a \( p \times 1 \) random vector with distribution function \( F \). We define the distortion associated with \( g \) clusters as

\[
d_g(\Gamma, \Sigma, F) = \inf_{\mathbf{m}_1, \ldots, \mathbf{m}_g \in \mathbb{R}^p} \{ E[r(\mathbf{X}, \mathbf{M}, \Gamma, \Sigma)] \}.
\]

(2.8)

For any \( \Gamma, \Sigma \) and \( F \),
1. \(d_g(\Gamma, \Sigma, F) \geq 0, \ \forall g;\)

2. \(d_1(\Gamma, \Sigma, F) \geq d_2(\Gamma, \Sigma, F) \geq \cdots \geq d_g(\Gamma, \Sigma, F) \geq \cdots .\)

For notational convenience we shall sometimes refer to \(d_g(\Gamma, \Sigma, F)\) simply as \(d_g.\)

Note that, given \(\Gamma\) and \(\Sigma,\) the estimated distortion is \(\hat{d}_g(\Gamma, \Sigma, F) = d_g(\Gamma, \Sigma, \hat{F}),\) where \(\hat{F}\) is the empirical distribution based on the sample \(\mathcal{X} = (X_1^T, \ldots, X_n^T)^T.\) That is,

\[
\hat{d}_g(\Gamma, \Sigma, F) = d_g(\Gamma, \Sigma, \hat{F}) = \inf_{m_1, \ldots, m_g \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} (X_i - o_i)^T \Sigma^{-1} (X_i - o_i) \right\}. \tag{2.9}
\]

where \(o_i = c(X_i, M, \Gamma).\)

The interpretation of the estimated distortion function for the special case where \(\Gamma = \Sigma = I_p,\) the \(p\)-dimensional identity matrix is simple. Let \(c_j = \sum_{j=1}^{g} m_j \mathbb{I}_j(X_i),\) where

\[
\mathbb{I}_j(X_i) = \mathbb{I}\{ (X_i - m_j)^T \Gamma^{-1} (X_i - m_j) = \min_{1 \leq s \leq g} [(X_i - m_s)^T \Gamma^{-1} (X_i - m_s)] \}.\]

Then, for a given subset \(\mathcal{X}_j\) of \(\mathcal{X},\) the vector \(c_j\) is the best representative of the observations in \(\mathcal{X}_j\) in the sense that it minimizes the sum of the squared lengths of the “error” vectors \(X_i - c_j,\) where \(X_i \in \mathcal{X}_j.\) The estimated distortion is thus a measure of the mean squared error incurred in representing \(\mathcal{X}\) by \(g\) cluster centers \(c_1, c_2, \ldots, c_g.\) Intuitively, the estimated distortion provides a reasonable objective function with small values indicating good partitions.
Note that, for a data set of size \(n\), \(n\hat{d}_p(I_p, I_p, F)\) is the so called sum-of-squared-error criterion. This is sometimes simply referred to as the trace criterion. The trace criterion can therefore be viewed as a special case of the estimated distortion.

Another commonly used objective function is the determinant criterion (Friedman and Rubin, 1967). This criterion minimizes the determinant of the pooled within groups sum of squares and product matrix, \(W\). We should note that partitions induced by minimizing the trace or determinant criteria are not only heuristically motivated, but also correspond to the maximum likelihood estimates of well-defined statistical models. Thus, we expect these methods to work reasonably well provided the data set “is a realization of random variables obeying the underlying statistical laws” (Gallegos and Ritter, 2005). For instance, the trace criterion tends to be optimal when the subpopulations in the data set are all normally distributed with unknown mean vectors and the same spherical covariance matrices \(\sigma^2 I_p\). For the determinant criterion, the covariance matrices must be equal but not necessarily spherical. As such, the determinant criterion tends to be invariant not only with respect to location but also with respect to scale transformation. These types of clustering are often referred to as minimum variance partitions. It is important to note that regardless of which objective function is used, the computational technique for searching among available partitions is a crucial part of the solution to the first question.

Except in small sample cases, the evaluation of objective functions for all possible partitions is most of the time computationally impossible, and thus making restricted searches inevitable. For instance, as noted by Duda and Hart (1973), the
number of ways to partition a set of $n$ objects into $g$ disjoint non-void subsets is given by

$$J(n, g) = \frac{1}{g!} \sum_{i=0}^{g} \binom{n}{i} (-1)^i (g - i)^n.$$  

We note that the function $J(n, g)$ grows exponentially fast in $n$. Even for small sets of objects the number of possible partitions into $g$ disjoint subsets can be staggering. For example, for $n = 100$ objects there are $J(100, 2) \approx 10^{30}$ ways to partition them into two subsets. It can be shown that the problem of computing a set of $g$ clusters of high quality is NP-complete. By high quality clusters, we mean clusters with minimal membership error. A problem is said to be NP-hard if an algorithm for solving it can be translated into one for solving any other NP-problem (solvable in nondeterministic polynomial time). A problem which is both NP and NP-hard is called an NP-complete problem.

A common approach is to use an iterative relocation algorithm. An iterative relocation algorithm often starts with an initial arbitrary partition and attempts to improve the value of the objective function by moving samples from one group to the other. It should be noted, however, that the convergence to a global optimum of these approaches is often not guaranteed. The worth of a final solution may be assessed by repeating the computation several times for different initial partitions and investigating the stability of the derived solution.

A plot of the estimated “distortion” $\hat{d}_g$ versus cluster number $g$ is a natural and simple way of finding the “optimal” cluster number in a data set. Suppose a data
set of size $n$ contains $g_0$ natural clusters. As noted by Duda and Hart (1973), one will generally expect $d_g$, the distortion associated with fitting $g$ clusters to the above data set to decrease rapidly until $g = g_0$, after which this decrease is much slower until it reaches zero at $g = n$. That is, the distortion curve is generally expected to level off whenever $g$ exceeds the “true” number of clusters $g_0$, in the data set. However in many cases this leveling is often not pronounced, thus making it difficult to judge the exact value of $g$ at which the leveling actually starts (the estimated “true” cluster number). We illustrate this problem in the following example.

**Example 2.2.1** We consider a simulated data set containing three Gaussian clusters each having the identity covariance structure and containing 50 points embedded in two
Figure 2.2: The estimated Jump curve $\hat{J}_g(p/2)$ for the simulated data set in Figure 2.1(a).

dimensions. The centers of these clusters are respectively

$$c_1 = (0, 0)^T, \ c_2 = (-2, -2)^T, \ c_3 = (2.5, -2.5)^T.$$ 

Figure 2.1(a) depicts an instance of a simulated data set described in Example 2.2.1. Figure 2.1(b) is the corresponding estimated distortion curve. Based on this estimated distortion curve, it is not entirely obvious where the leveling starts (the estimated “true” cluster number).
2.3 The Jump Method

As an alternative to the distortion curve, Sugar and James (2003) proposed the so called jump method. For a \( p \)-dimensional data set, the jump method estimates the cluster number by

\[
\arg \max_g \hat{J}_g(p/2),
\]

where

\[
J_g(p/2) = \left( \frac{d_g}{p} \right)^{-p/2} - \left( \frac{d_{g-1}}{p} \right)^{-p/2}, \quad g = 1, \ldots, g_{\text{max}},
\]

is called the jump (associated with fitting \( g \) clusters to a data set) and \( d_g \) is the “distortion,” with \( \{d_0\}^{-p/2} \equiv 0 \). For convenience, Sugar and James (2003) let \( \Sigma = \Gamma = I_p \) in implementing their method. Figure 2.2 shows an estimated jump curve \( \hat{J}_g(p/2) \) for the simulated data set from Example 2.2.1. The maximum of \( \hat{J}_g(p/2) \) is clearly at \( g = 3 \) clusters.

2.3.1 Motivation of the Jump Method: Rate Distortion Theory

The jump method derives it’s motivation from a branch of information theory known as rate distortion theory. Rate distortion theory attempts to determine the minimum amount of entropy or information that can be transmitted from a source (input signal) to a “receiver” (output signal) with a specified distortion, \( D \). The transmission of perfect signals, which are typically a sequence of realizations of continuous variables, will require the storage of an infinite number of bits. This is not feasible. Thus, in practice a finite number of representatives are often used to approximate the output signal as
accurately as possible. Using a finite number of representatives to approximate the output signal introduces some errors or “distortion” in the original input signal.

For a given number of representatives, the rate distortion problem then is to determine the minimum achievable distortion. Rate $R$ is usually understood as the number of bits per data sample to be stored or transmitted. For any given distortion $D$, the minimum rate achievable is often referred to as the rate distortion function $R(D)$, and the corresponding minimum achievable distortion for a given rate is referred to as the minimum achievable distortion $D(R)$. Asymptotic results show that both $D(R)$ and $R(D)$ are convex nondecreasing functions of $R$ and $D$, respectively.

Cluster analysis typically attempts to summarize a population in the best possible way by segmenting the population into a fixed number of homogeneous natural groups or clusters, $g$. The cluster centers provide canonical representation of their cluster membership. This can be viewed as performing data compression on the population. This compression naturally introduces an imprecision, or a distortion in the summarized population. For a fixed cluster number $g$, the minimum achievable distortion associated with fitting $g$ centers to a data set $d_g$ provides a measure of the “best” summary that can be achieved by grouping the population into $g$ clusters. The assumption here is that the best possible summary will be achieved when the appropriate $g$ is fit to the data set.

Cluster analysis may thus be viewed as a rate distortion problem. Here, the number of cluster centers corresponds to the fixed number of representatives used in rate distortion theory. $d_g$ serves as a measure of distortion in the sense of $D(R)$ in rate
distortion theory. Empirical evidence suggests that $d_g$ is indeed similar to $D(R)$. In fact $d_g \rightarrow D(\frac{1}{p} \log_2 g)$ as $p \rightarrow \infty$. Sugar and James (2003) capitalize on the general shape of $D(R)$ to motivate their method. For simple cases, the rate $R$ is related to the fixed number of representatives (clusters) $g$, by $g = 2^R$.

Under squared error, the distortion rate function for a $p$-dimensional normal variable with mean $\mu$ and covariance structure $\sigma^2 I_p$ is known to be

$$D(R) = p\sigma^2 2^{-2R}.$$  

Below is a key result extracted from Sugar and James(2003):

**Theorem 2.3.1** Let $X$ have the distribution of a mixture of $g_0$ Gaussian clusters with equal priors and common covariance $\Sigma$. Let $\Delta \sqrt{p}$ be the minimum Euclidean distance between cluster means after standardizing the space by multiplying by $\Sigma^{-1/2}$. Then for $g < g_0$

$$\lim_{p \rightarrow \infty} d_g = \infty$$

provided $\Delta$ is bounded away from zero. Furthermore, for $k^p - 1 \leq g \leq k^p$,

$$\lim_{p \rightarrow \infty} d_g = k^{-2}$$

provided $\Delta > 6$. 

27
Sugar and James (2003) deduced the following from the above theorem,

\[
d_{g}^{-p/2} \approx \begin{cases} 
    a \frac{2}{g_0}, & g \geq g_0 \\
    0, & g < g_0,
\end{cases}
\]  

(2.13)

where \(0 < a < 1\). Equation (2.13) suggest that the largest jump occurs at \(g_0\), the “true” cluster number. The above findings thus provide a theoretical basis for their proposed method. Inferring from equations (2.12) and (2.13), the jump function may be approximated by

\[
J_{g}(p/2) \approx \begin{cases} 
    0, & g < g_0, \\
    a, & g = g_0, \\
    \frac{a}{g_0}, & g > g_0.
\end{cases}
\]  

(2.14)

Note that by (2.14), the largest “jump” clearly occurs exactly at \(g = g_0\), the “true” number of clusters.

Equation (2.14) suggests that with the “appropriate” transformation power \(p/2\), this method is expected to exhibit a sharp jump at the “true” number of cluster components \(g_0\). Asymptotic results on mixtures of non-Gaussian clusters are provided by Sugar and James (2003).

2.3.2 Some Comments on the Method of Sugar and James (2003)

The proposed method of Sugar and James (2003) performs well in cases where the clusters are of the same size, with little or no correlations between features/dimensions. However, even for well separated clusters, we note that the method of Sugar and James (2003) performs poorly in predicting the correct number of “true” clusters under the
following circumstances:

- when there appear to be high correlations between the variables;
- when the number of features/dimension is large;
- in the presence of noise dimensions.

**Example 2.1.2** We consider a simulated bivariate sample generated to possess three Gaussian clusters each containing 50 points. The centers and covariances of these clusters are respectively

\[ c_1 = (0, 0)^T, \quad c_2 = (-2, -2)^T, \quad c_3 = (2.5, -2.5)^T, \]

and

\[ \Sigma_1 = \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}, \quad \Sigma_3 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}. \]

An instance of the data generating model in the example above is depicted in Figure 2.3(a). For the simulated data set shown in Figure 2.3(a), the method of Sugar and James (2003) failed to detect the correct number of clusters. Figure 2.3(c) depicts the “jump” curve (A plot of the estimated jump function, \( \hat{J}_g(p/2) \), with \( p = 2 \) versus the number of clusters, \( g \)) corresponding to the simulated data set depicted in Figure 2.3(a). It estimates the cluster number as six, when the true cluster number is three. This instance demonstrates the inability of the jump method to capture the
Figure 2.3: Simulated data, estimated distortion and the estimated Jump curve \( \hat{J}(p/2) \) curve from Example 2.3.2: (a) Simulated data containing \( g_0 = 3 \) elongated clusters; (b) Estimated distortion function \( \hat{d}_g \); (c) Estimated jump function \( \hat{J}_g(p/2) \) chose \( \hat{g} = 6 \).
“true” number of clusters for cases where there exist correlations between dimensions.

For instances in which correlations exist between the variables, Sugar and James (2003) assert that some value $0 < t < p/2$ may be superior to $p/2$ in predicting the optimal cluster number. They interpret $2t$ as the “effective” number of dimensions. They therefore generalize their idea by suggesting that, given the “appropriate” transformation power $t > 0$, the largest value in the jump function

$$J_g(t) = \left(\frac{d_g}{p}\right)^{-t} - \left(\frac{d_{g-1}}{p}\right)^{-t}, \quad g = 1, \ldots, g_{\text{max}},$$

will occur at $g = g_0$ the “true” cluster number provided there is enough separation between the clusters. However finding the “appropriate” $t$ is non-trivial.

Sugar and James (2003) used principal component analysis to reduce the dimension of the well known Iris data from four to two (the “effective” dimension). With two as the “effective” dimension, they successfully predicted the true number of clusters ($g_0 = 3$) in the Iris data set.

It should be noted, however, that even though principal components are often used for dimension reduction, as cautioned by Chang(1983), in some instances transforming the data into principal components may obscure rather than reveal the groups of interest.

The basic idea of the jump method is that, given the “appropriate” transformation power $t > 0$, the largest value in the jump function $J_g(t)$ will occur at $g = g_0$, the “true” cluster number provided there is enough separation between the clusters. It
therefore makes sense to estimate the “optimal” cluster number as the value of \( g \) that provides the largest value in \( J_g(t) \). This idea forms the basis of our proposed methods. We shall refer to our approach as the maximal jump approach.

### 2.4 The Maximal Jump Approach

Our approach arose from the demonstrated promise of the jump method of Sugar and James (2003).

**Definition 2.4.1** Let \( c > 0 \) and \( d_g \) be the distortion associated with a distribution \( F \).

The jump function \( J \),

\[
J : \mathbb{N}^+ \times \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow [0, \infty),
\]

is defined as

\[
J(g, t, c) = \left( \frac{d_g}{c} \right)^{-t} - \left( \frac{d_{g-1}}{c} \right)^{-t}, \; t \in (0, \infty), \; g = 1, \ldots, g_{\max}, \quad (2.15)
\]

with \( \{d_0\}^{-t} \equiv 0 \) the jump function.

\( J(g, t, c) \) may be viewed as the excess transformed distortion obtained by fitting an extra cluster to a data set. By transformed distortion we mean the quantity \( (d_g/c)^{-t} \). For a given \( c \), large values of \( J(g, t, c) \) suggest optimality at \( g \) and \( t \). The constant \( c \) has some practical interpretation. Suppose \( d_g \) is the distortion function derived from some \( n \times p \) data set \( \mathcal{X} \). It can easily be shown that the distortion function derived from \( \mathcal{X}/\sqrt{c} \) is \( d_g/c \). Thus, by dividing \( d_g \) by \( c \), we are indirectly transforming the data set \( \mathcal{X} \) by...
dividing each of its elements by $\sqrt{c}$.

**Proposition 2.4.1** Define $d_0^{-t} \equiv 0$. Let $c > 0$ and consider the jump function

$$
J(g, t, c) = \left(\frac{d_g}{c}\right)^{-t} - \left(\frac{d_{g-1}}{c}\right)^{-t}, \quad t \in (0, \infty), \; g = 1, \ldots, g_{\text{max}},
$$

(2.16)

where $d_g$ is the distortion for $g$ clusters. If $d_g > c$ for all $g$, then,

$$
\tau_g = \frac{\ln \left(\frac{\ln d_g - \ln c}{\ln d_{g-1} - \ln c}\right)}{\ln \left(\frac{d_g}{d_{g-1}}\right)},
$$

satisfies

$$
J(g, t, c) \leq J(g, \tau_g, c), \; \text{for all } t \in (0, \infty).
$$

Furthermore, if in addition, $d_1 > d_2 > \cdots > d_{g_{\text{max}}}$, then

(a) $\lim_{d_g \to c} \tau_g = \infty$

(b) $\tau_2 < \cdots < \tau_{g_{\text{max}}}$

(c) $0 < \tau_g < 1$ for all $g$ provided $c^{-1}d_g > \exp(1)$ for all $g$.

The proof of Proposition 2.4.1 is given in Appendix A.

**Definition 2.4.2** For a given $c > 0$, let $\tau_g \in (0, \infty)$ such that

$$
J(g, t, c) \leq J(g, \tau_g, c), \; \text{for all } t \in (0, \infty).
$$

We define the maximal jump function

$$
H : \mathbb{N}^+ \times \mathbb{R}^+ \to [0, \infty),
$$
by
\[ H(g, c) = J(g, \tau_g, c), \text{ for } g = 2, \ldots, g_{\text{max}}. \]

For \( g = 2, \ldots, g_{\text{max}} \), we define the following:

- \( W(g, c) = H(g, c) - H(g + 1, c) \)
- \( U(g) = \lim_{c \to 0} (-\ln c) [W(g, c)] \)

In Appendix B we show
\[
U(g) = \exp(-1) \left[ \ln \left( \frac{d_g - 1}{d_g} \right) - \ln \left( \frac{d_g}{d_{g+1}} \right) \right]. \tag{2.17}
\]

We propose three estimators of the cluster number \( g \) as follows:

**Method 1**: \( \hat{g} = \arg\max_g \hat{H}(g, c) \);

**Method 2**: \( \hat{g} = \arg\max_g \hat{W}(g, c) \);

**Method 3**: \( \hat{g} = \arg\max_g \hat{U}(g) \).

Note that **Method 1** and **Method 2** require the pre-specification of \( c \). In general, \( 0 < c < d_{g_{\text{max}}} \) as \( H(g, c) \) tends to be undefined for some \( g \)'s whenever \( c > d_{g_{\text{max}}} \).

We also note the following:

(a) \( 0 \leq H(g, c) \leq 1 \),

(b) \( \lim_{d_g \to c} H(g, c) = 1 \).

Note that (b) is a direct consequence of Proposition 2.4.1(a). Inferring from (b), the maximal jump function \( H(g, c) \) has the tendency to grow arbitrarily large for any \( g \) for which \( d_g \) is approximately close to \( c \). Therefore, one can easily overestimate the “true”
cluster number using **Method 1** or **Method 2**.

To prevent this from happening, in implementing **Method 1** or **Method 2**, it is important that $c$ be chosen “sufficiently” less than $d_{g_{\text{max}}}$.

This implies that the “optimal” $c$ tends to vary from data set to data set. Throughout this dissertation we shall estimate $c$ by $\hat{c} = \hat{d}_{2g_{\text{max}}}$. This estimate clearly lies between 0 and $\hat{d}_{g_{\text{max}}}$ and tends to produce good results.

**Method 1** is based directly on the maximal jump function $H(g, c)$, and hence we shall refer to it as the maximal jump method or simply as $MJ(\hat{c})$. Also **Methods 2** and **3** are based on the successive differences of the maximal jump function $H(g, c)$, evaluated at $\hat{c}$ and 0 respectively. Hence we shall refer to them respectively as $MJD(\hat{c})$ and $MJD(0)$ or collectively as the maximal jump differences methods, $MJD$.

Note that **Method 1** follows directly from the Sugar and James (2003) idea that there exists some “appropriate” or optimal transformation power $t$, for which the largest value in the jump function, $J(g, t, c)$, tends to occur at the “true” cluster number $g_0$.

All three proposed methods are easy to implement. For example, to implement $MJ(\hat{c})$ we proceed as follows: First, set $\hat{c} = \hat{d}_{2g_{\text{max}}}$ and for each $g \in \{2, \ldots, g_{\text{max}}\}$, find the transformation power $t$ that provides the maximal value in $J(g, t, \hat{c})$. Denote this transformation power by $\tau_g$. That is $\tau_g = \text{argmax}_{t > 0} J(g, t, \hat{c})$. Next, fix $t = \tau_g$ in $J(g, t, \hat{c})$ and choose $g \in \{2, \ldots, g_{\text{max}}\}$ to maximize $\hat{J}(g, \tau_g, \hat{c})$ to obtain $\hat{g}$. Thus the pair $(g, \tau_g)$ that maximizes $\hat{H}(g, \hat{c}) = \hat{J}(g, \tau_g, \hat{c})$ are our “optimal” cluster number and
“appropriate” transformation respectively.

We outline the $MJ(\hat{c})$ approach for determining the “optimal” cluster number in data as follows:

1. For $g = 1, \ldots, g_{\text{max}}$, compute the distortion $\hat{d}_g = d_g(I_p, I_p, \hat{F})$, where $\hat{F}$ is the empirical distribution based on the sample $X_1, X_2, \ldots, X_n$.

2. For some $0 < \hat{c} < \hat{d}_{g_{\text{max}}}$ and for $g = 2, \ldots, g_{\text{max}}$, compute

$$\hat{H}(g, \hat{c}) = \hat{J}(g, \hat{\tau}_g, \hat{c}) = \left(\frac{\hat{d}_g}{\hat{c}}\right)^{-\hat{\tau}_g} - \left(\frac{\hat{d}_{g-1}}{\hat{c}}\right)^{-\hat{\tau}_g}$$

(2.18)

(2.19)

$(\hat{c} = \hat{d}_1$, where $I = 2g_{\text{max}}$ works well in practice).

3. Estimate the “optimal” cluster number $g_0$, by

$$\hat{g} = \arg\max_g \hat{H}(g, \hat{c}).$$

(2.20)

Figure 2.4 shows the estimates of the cluster number for Examples 2.2.1 and 2.3.2 using our proposed methods. As indicated by the peaks of these plots, all three proposed methods correctly predicted the true cluster number.
Figure 2.4: (a) Plot of $\hat{H}(g, \hat{c})$ versus Cluster number $g$ for Example 2.2.1 (b) Plot of $\hat{H}(g, \hat{c})$ versus Cluster number $g$ for Example 2.3.2 (c) Plot of $\hat{W}(g, \hat{c})$ versus Cluster number $g$ for Example 2.2.1 (d) Plot of $\hat{W}(g, \hat{c})$ versus Cluster number, $g$ for Example 2.3.2 (e) Plot of $\hat{U}(g)$ versus Cluster number $g$ for Example 2.2.1 (f) Plot of $\hat{U}(g)$ versus Cluster number $g$ for Example 2.3.2
2.4.1 Computational Strategy: k-means

The clustering technique to be used as an intermediate partition engine in determining the best subdivision of $\mathcal{X}$ for a given number $g$ of clusters is crucial. For convenience, we adopt the so called k-means clustering algorithm, also historically known as Forgy’s Method or Hartigan’s algorithm as our partition engine. It is therefore important that we address some important issues in its implementation.

The k-means algorithm objective function is

$$J = \min_{c_1, \ldots, c_g} \left\{ \sum_{j=1}^{g} \sum_{X_i \in \mathcal{X}_j} S^2(X_i, c_j) \right\},$$

where $S^2(X_i, c_j) = S^2_{ji}$ represents the squared distance of the data point $X_i \in \mathcal{X}_j$ to the cluster center $c_j$, associated with $\mathcal{X}_j$. Given $g$ clusters, k-means estimates the $j$th cluster by

$$\hat{\mathcal{X}}_j = \left\{ X_i \in \mathcal{X} | S^2_{ji} = \min_{k=1,\ldots,g} S^2_{jk} \right\}. \tag{2.21}$$

The center of the $j$th cluster, $c_j$, is derived by setting $\partial J / \partial c_j = 0$. For cases where $S^2_{ji}$ is the squared Euclidean distance $S^2_{ji} = ||X_i - c_j||^2$, the k-means cluster center estimate is given by

$$\hat{c}_j = \frac{1}{n_j} \sum_{X_i \in \hat{\mathcal{X}}_j} X_i, \tag{2.22}$$

where $n_j$ is the number of points in $\hat{\mathcal{X}}_j$.

The k-means algorithm consists of alternating updates of centers (2.22) and
the partition using (2.21), until convergence of a fixed number of iterations.

The k-means algorithm essentially finds locally optimal solutions minimizing the sum of the squared $L_2$ distance between each data point and it’s nearest cluster center. Several variants of the k-means algorithm exist in the literature, but common to all these variants is an iterative scheme that attempts to improve the estimate of the mean of each cluster, and then to re-classify each data point to the cluster with nearest mean.

In implementing this algorithm one needs to pay attention to the following issues:

1. Initialization of the algorithm: Since the basic k-means algorithm, like all hill climbing strategies, does not guarantee finding a global optimum and only finds a local minima, the solution obtained often depends very much on how it is initialized. There are a number of different methods used to initialize the algorithm. A common and effective initialization method is to divide the data set into $g$ clusters at random (Pena et al., 1999). Forgy (1965) proposed an initialization method based on choosing $g$ points of the data set at random to represent $g$ centroids, and the rest of the points are assigned to the cluster represented by the nearest centroid. Other initialization methods haven been discussed in the literature. These include ones proposed by MacQueen (1967), Kaufman and Rousseeuw (1990) and more recently by Bradley and Fayyad (1998).

2. Distance measure: The algorithm spends most of its time computing the distance.
So, the measure chosen will have significant impact on the total time taken to compute. The three most common distance measures used are -

- **L1 distance**: This is the sum of the absolute values of the difference in each component. This is the simplest to compute with complexity of $O(n)$ and is robust to outliers. This is often used in image processing applications.

- **L2 (Euclidean) distance**: This is the most popular distance measure, and is the square root of sum of the squares of the difference in each component. Since distances are compared with each other, the square root can be omitted. This is computationally $O(n)$ but a bit more expensive than L1. It is more elegant than L1 as it computes the radial distance from the centroid.

- **Mahalanobis distance**: This distance is similar to the L2 distance, except the components are weighted by the inverse of the covariance of the cluster from which the distance is being computed. Computationally it is $O(n^2)$.

3. Number of iterations: Theoretically, k-means should terminate when no more data points are changing classes. There are proofs of termination for k-means that rely on the fact that both steps of k-means (assign data points to nearest centers, move centers to cluster centroids) reduce variance. So eventually, there is no move to make that will continue to reduce the variance. However, in practice that may require a very large number of iterations. Therefore, typically, two criteria are used:

- terminate after a predefined fixed number of iterations; or
- terminate after fewer than $n$ sample change classes.
4. Dead clusters: Sometimes a centroid may not have any members in the next iteration. Such a cluster needs to be removed.

2.5 Model Based Clustering

Model-based clustering techniques have been widely used and have shown promising results in many applications involving complex data. These techniques generally assume that the observed data come from a source with several subpopulations, instead of just a single population, and hence proceed to model the overall population as a mixture of these subpopulations via finite mixture models.

Model-based clustering can be traced as far back as Wolf (1963). For a review of this technique, see Peel and McLachlan (2000) and Fraley and Raftery (1998). The general form of a finite mixture density with $g$ subpopulations or clusters is

$$f(x; \theta) = \sum_{j=1}^{g} \pi_j f_j(x; \theta), \quad (2.23)$$

where the $\pi_j$’s are called the mixing proportions and represent the probabilities for each subpopulation or group $j$. The $f_j(\cdot)$’s are densities of the subpopulations.

In practice, with continuous data, each of these subpopulations or clusters is assumed to be generated from the multivariate normal distribution. This may be partly because of computational convenience. Data sets containing observations with longer tails or atypical observations may unduly affect the fit of the mixture model if the multivariate normal distribution is used to represent these subpopulations. McLachlan
and Basford (1998) provide a more robust approach that fits a mixture of $t$-distributions to alleviate this problem.

Model-based clustering methods have the advantage of applying well studied model selection techniques in deciding the appropriate number of mixture components. In particular, the mixture-model approach allows the use of approximate Bayes factors (see Kass and Raftery (1995)) for comparing different models. Using a heuristic approach, Banfield and Raftery (1993) derived an approximation to twice the logarithm of the Bayes factor, called AWE for Average Weighting Evidence, for determining the number of components. Fraley and Raftery (1998) used the Expectation-Maximization (EM) algorithm to find the mixture maximum likelihood estimator and obtained a more reliable approximation of twice the logarithm of the Bayes factors as the Bayesian Information Criterion, BIC. They then propose using BIC in selecting not only the appropriate number of components in the mixture models but also for selecting the underlying covariance structure of each component. This approach can easily be implemented using their MCLUST software.

Banfield and Raftery (1993) provide a parametrization of the covariance matrix in terms of its eigenvalue decomposition given by

$$
\Sigma_j = \lambda_j D_j A_j D_j^T,
$$

where $\lambda_j$ is a scalar, $D_j$ is the orthogonal matrix of eigenvectors, and $A_j$ is a diagonal matrix whose elements are proportional to the eigenvalues of $\Sigma_j$. Geometrically, $D_j$ determines the orientation of the principal components of $\Sigma_j$, $A_j$ determines the shape.
Table 2.2: Parameterizations of the covariance matrix $\Sigma_j$ and their geometric interpretations.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\Sigma_j$</th>
<th>Distribution</th>
<th>Volume</th>
<th>Shape</th>
<th>Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\lambda I_p$</td>
<td>Spherical</td>
<td>Equal</td>
<td>Equal</td>
<td>NA</td>
</tr>
<tr>
<td>2</td>
<td>$\lambda I_p$</td>
<td>Spherical</td>
<td>Variable</td>
<td>Equal</td>
<td>NA</td>
</tr>
<tr>
<td>3</td>
<td>$\lambda DAD$</td>
<td>Ellipsoidal</td>
<td>Equal</td>
<td>Equal</td>
<td>Equal</td>
</tr>
<tr>
<td>4</td>
<td>$\lambda_D J_A D_j$</td>
<td>Ellipsoidal</td>
<td>Variable</td>
<td>Variable</td>
<td>Variable</td>
</tr>
<tr>
<td>5</td>
<td>$\lambda D_A D_j$</td>
<td>Ellipsoidal</td>
<td>Equal</td>
<td>Equal</td>
<td>Variable</td>
</tr>
<tr>
<td>6</td>
<td>$\lambda D_A D_j$</td>
<td>Ellipsoidal</td>
<td>Variable</td>
<td>Equal</td>
<td>Variable</td>
</tr>
</tbody>
</table>

of the clusters, and the volume of the corresponding ellipsoid is determined by $\lambda_j$. Thus, the above parametrization provides a sufficiently flexible class of models to accommodate the wide variety of characteristics often seen in data. Six special cases of the above parametrization and their effect on the orientation, shape, volume and distribution of clusters are shown in Table 2.2.

In practice, a plot of BIC versus number of mixture components $g$, is used in selecting the “true” number of clusters. Usually, BIC differences of less than 2 correspond to weak evidence, differences between 2 and 6, are an indication of a positive evidence, differences between 6 and 10 support strong evidence, and differences greater than 10 indicate very strong evidence (Jeffreys, 1961; and Kass and Raftery, 1995).

It is worth mentioning that the MCLUST software initializes the mixture maximum likelihood via model-based hierarchical agglomerative clustering. Model-based hierarchical agglomerative clustering is essentially a procedure that computes an approximate maximum for the classification likelihood,

$$L_{cl}(\theta_1, \ldots, \theta_g; l_1, \ldots, l_n) = \prod_{i=1}^{n} f_{l_i}(X_i|\theta_{l_i}), \quad (2.24)$$

43
where the $l_i$'s are labels indicating a unique classification of each observation, $l_i = j$ if $X_i$ belongs to the $j$th component. They tend to successively merge pairs corresponding to the greatest increase in the classification likelihood among all possible pairs. In the absence of labels, the procedure treats each observation as a singleton cluster.

Leroux (1992) showed that using BIC to select models will not underestimate the number of groups asymptotically. Keribin (1998) also showed that the BIC is consistent for the number of components. In our simulation study, the estimate of the number of clusters in a data set is taken to be the value of $g$ that produces the largest BIC among all the above mentioned six models.

It is noteworthy that, even though regularity conditions are not fully met for mixture models, in the clustering context there is some considerable practical as well as theoretical support for its use (Dasgupta and Raftery, 1998). Also, Cadez and Smyth (2001) present a result of “almost” concavity of the log-likelihood of a data set under a $g$-component mixture as a function of the number of components $g$. They show that the first-order Taylor approximation of the log-likelihood as a function of $g$ is concave under general conditions. Hence a penalized log-likelihood model selection criterion based on a penalty term that is concave or linear in $g$ is “almost” concave. This implies that if the “almost” concave turns out to be concave, we would expect only one peak in the penalized log-likelihood, and this peak can easily be identified.

Even though clustering methods based on the multivariate normal distribution have been successful in a number of applications, their practical use for non-Gaussian data tends to be limited without some modifications. Also their usefulness tends to be
limited for both high-dimensional and large data sets (Fraley and Raftery, 1998).

Note that for these methods, the number of parameters per component in the multivariate normal mixtures grows as the square of the dimension. Also for cases where the dimension is larger than the sample size, the covariance estimates in the model will most of the time be singular, thereby causing the EM algorithm to break down. We should, however, note that the more parsimonious Models 1 and 2 in Table 2.2 may still be applicable.
Chapter 3

Simulation Study

The ability of a clustering algorithm to recover “natural” clusters embedded in data depends on several factors. For instance, Milligan and Cooper (1985) stated the importance of the number of clusters \( g \) present in the data set. More often than not, a data set with a large number of dimensions contains more information about the natural clustering present in the data than one with a small number of dimensions. Thus, as remarked by Krzanowski and Lai (1988), a good clustering criterion can easily capitalize on this information to exhibit a greater precision with increased dimensionality, while a poor criterion will tend to display a fairly constant criterion irrespective of dimensionality. However, it is also true that a large number of noise dimensions could lead to poor results. By noise dimensions, we mean uninformative variables. Cluster size (number of points per cluster) has been found by previous researchers to have a significant effect on cluster recovery (Gordon, 1981).

The use of simulated data as an aid to validating and comparing clustering
methods has been highly advocated in the clustering literature. Milligan and Cooper (1985) provide a Monte Carlo study of 30 procedures for determining the number of clusters in artificial data sets containing varying number of non-overlapping clusters. They found the method suggested by Calinski and Harabasz (1974) to be best in terms of cluster number recovery. Sugar and James (2003) also compare their jump method for determining the number of clusters in data set to methods suggested by Calinski and Harabasz (1974), Hartigan (1975), Krzanowski and Lai (1985), Kaufman and Rousseeuw (1990) and the \textit{GAP} method of Tibshirani et al. (2001) in a Monte Carlo study. They conclude that their method outperformed the other methods in a wide variety of situations.

In most applications, the underlying true structure of the data is often hidden in various types of errors or noise, making it difficult for clustering methods to recover the true structure. The robustness of clustering methods to the presence of noise dimensions in data is a desirable property. Validation research is necessary since there is no guarantee that a clustering algorithm would be able to uncover the underlying cluster structure even under ideal situations.

In Section 3.1 through to Section 3.4, we study the performance of our three proposed methods by comparing them to the jump method of Sugar and James (2003) and to the model-based approach of Fraley and Raftery (1998) via Monte Carlo simulations.
3.1 Simulation 1: When No Noise Features are Present

The purpose of this simulation study is to ascertain the performance of our proposed methods in terms of their cluster number recovery ability by comparing them to the jump method of Sugar and James (2003) and to the model-based approach of Fraley and Raftery (1998) when no noise dimensions are present in the data. In addition, we are interested in studying the performance of these methods as the separation between cluster centers increases.

3.1.1 Simulation 1 Design

The simulation design of the data generating model is outlined below:

1. The “true” number of clusters, \( g_0 \).
   
   This is randomly generated from the discrete uniform distribution, \( DU\{2, \ldots, 5\} \).

2. The dimension of \( \mathcal{X} \), \( p \).

   The number of dimensions \( p \), is randomly generated from the discrete uniform distribution, \( DU\{2, \ldots, 10\} \).

3. The cluster centers, \( M^* \).

   For a fixed cluster number \( g_0 \) and a fixed dimension \( p \), three levels of cluster centers represented by the \( g_0 \times p \) matrices \( M_1^* \), \( M_2^* \) and \( M_3^* \) were considered:

   The elements of the matrix \( M_a^* \), \( a = 1, 2, 3 \) were generated randomly from the \( N(0, a^2) \) distribution.
4. Sample size, $n$.

The sample size $n$ is randomly generated from the discrete uniform distribution, $DU[\{100, 101, \ldots, 400\}]$.

5. Cluster size

By cluster size, we mean the number of points per cluster. Let $n_j$, $j = 1, \ldots, g_0$, represent the number of points in the $j$th cluster, and let the set $N_{g_0} = \{(n_1, \ldots, n_{g_0}) : \sum_{j=1}^{g_0} n_j = n\}$ be the collection of the number of points within each cluster. Conditioned on $n$, we generate $N_{g_0}$ from the multinomial distribution with parameters $n$ and $\pi = (\pi_1, \ldots, \pi_{g_0})$, i.e.,

$$N_{g_0}|n \sim \text{Multi}(n; \pi_1, \ldots, \pi_{g_0})$$

where $\pi = (\pi_1, \ldots, \pi_{g_0})$ are themselves randomly generated from the Dirichlets distribution with parameters $a_{g_0} \mathbf{1}^T$, where $a_{g_0} > 0$ and $\mathbf{1}$ is a $g_0$-dimensional vector of ones, i.e,

$$\pi = (\pi_1, \ldots, \pi_{g_0}) \sim \text{Dir}(a_{g_0} \mathbf{1}^T).$$

The mean and variance of $\pi_j (j = 1, \ldots, g_0)$ are respectively

$$E(\pi_j) = \frac{1}{g_0} \quad (3.1)$$

and

$$\text{var}(\pi_j) = \frac{g_0 - 1}{g_0^2 (a_{g_0} g_0 + 1)}. \quad (3.2)$$
We note that, whereas the means of the \( \pi_j \)'s remains the same irrespective of the constant \( a_{g_0} > 0 \), the variances of the \( \pi_j \)'s get smaller as \( a_{g_0} \) gets larger. For a given \( g_0 \), we choose \( a_{g_0} \) such that \( \text{var}(\pi_j) = 1/16 \).

6. Covariance Structure

Let \( R_j \) \((j = 1, \ldots, g_0)\) be a \( p \times p \) matrix. We define \( R_j \) by

\[
R_j = [r_{jk}^{(j)}], \quad j = 1, \ldots, g_0, \quad i, k = 1, \ldots, p,
\]

where

\[
r_{jk}^{(j)} = \rho_j^{|i-k|}, \quad j = 1, \ldots, g_0, \quad i, k = 1, \ldots, p,
\]

is the \( i \)th row and \( k \)th column element of the \( p \times p \) matrix \( R_j \). That is,

\[
R_j = \begin{pmatrix}
1 & \rho_j & \ldots & \rho_j^{p-1} \\
\rho_j & 1 & \ldots & \rho_j^{p-2} \\
\vdots & \vdots & \ddots & \vdots \\
\rho_j^{p-1} & \rho_j^{p-2} & \ldots & 1
\end{pmatrix}, \quad (3.3)
\]

Let \( V_j \) \((j = 1, \ldots, g_0)\) be a \( p \times p \) diagonal matrix defined by

\[
V_j = \begin{pmatrix}
\sigma_{j1} & 0 & \ldots & 0 \\
0 & \sigma_{j2} & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \sigma_{jp}
\end{pmatrix}, \quad (3.4)
\]

where \( \sigma_{ji}, \ i = 1 \ldots, p \) is the standard deviation of the \( i \)th variate within the \( j \)th cluster. For a fixed \( p, \sigma_{j1}, \ldots, \sigma_{jp} \) are independently generated from the chi-square
distribution as follows

\[ \sigma_{j_1}, \ldots, \sigma_{j_p} | p \sim \chi_{10}^2 / 10. \]

Now, define the covariance structure of the \( j \)th cluster, \( \Sigma_j \), by

\[ \Sigma_j = V_j R_j V_j, \ j = 1, \ldots, g_0. \] (3.5)

For a fixed \( g_0 \), we generate \( \rho_1, \ldots, \rho_{g_0} \) from the discrete uniform distribution, \( DU\{[-0.5, 0, 0.5]\} \), that is

\[ \rho_1, \ldots, \rho_{g_0} | g_0 \sim DU\{[-0.5, 0, 0.5]\}. \]

We note that this formulation allows for different covariance structures within the different clusters.

Note that in the above simulation design, the separation between cluster centers is controlled at three levels. For each of these levels, a total of 200 data sets (consisting of 40 “blocks” each replicated five times) were randomly generated and clustered using all five clustering methods. Replicates within each “block” were randomly generated from a data generating model with the same set of model parameters. “Blocks” were randomly generated using a different set model parameters.
3.1.2 Simulation 1 Results

The application of a clustering method in recovering the “true” number of clusters embedded in a data set can result in two decisions, the correct one and a wrong one. Typically, two main types of decision error can result in the cluster number estimation process. The first error occurs when $\hat{g} > g_0$, where $g_0$ is the true number of clusters. The second error is for $\hat{g} < g_0$. The severity of these two types of errors depends purely on the context of the problem. Milligan and Cooper (1985) argue that the second type of error might be considered more serious in most applications since information could be lost by merging distinct clusters. However, how far a particular predicted cluster solution is away from the “true” number of clusters may also matter in most applications.

In view of the above, for each of the five methods, we present the frequency of the prediction error, $PE(\hat{g})$ of the estimated cluster number, $\hat{g}$ in Table 3.1. By prediction error we mean

$$PE(\hat{g}) = \hat{g} - g_0.$$ 

We also present the average value of $PE(\hat{g})$, $Bias(\hat{g}) = E(\hat{g}) - g_0$ and the root mean squared error, Root $MSE = \sqrt{E(\hat{g} - g_0)^2}$. For convenience, we shall sometimes refer to the Sugar and James (2003) method as the $SJ$ method, and to the Fraley and Raftery (1998) method as $FR$. From Table 3.1, we see that $SJ$ has the tendency to overestimate the cluster number by more than two. $FR$ appears to perform better than the other methods for cases where the separation between cluster centers is small in
terms of $P E(\hat{g})$. However as the separation between cluster centers increases, $FR$ and our three proposed methods tend to perform at an equal rate. Also note that when the separation between cluster centers is small ($M_i^*$), $FR$ and all three proposed methods tend to underestimate the “true” cluster number. Although not shown here, it is worth mentioning that when sampling from mixture densities, our proposed methods tend to predict the cluster number with excellent accuracy as $p$ becomes larger and larger, because the information for estimating $g_0$ is growing larger.

Table 3.1: Frequency of Prediction Error For the Five methods: $M_a^* \sim N(0,a^2)$, $a = 1,2,3$, $p \sim DU[\{2,\ldots,10\}]$, $p_I \sim DU[\{1,\ldots,p\}]$

<table>
<thead>
<tr>
<th>Cluster Center</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Overall Bias($\hat{g}$)</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\leq -2$</td>
<td>$-1$</td>
<td>$0$</td>
</tr>
<tr>
<td>$M_1^*$ SJ</td>
<td>0.05</td>
<td>0.04</td>
<td>0.06</td>
<td>0.04</td>
</tr>
<tr>
<td>$M_1^*$ FR</td>
<td>0.12</td>
<td>0.23</td>
<td>0.41</td>
<td>0.20</td>
</tr>
<tr>
<td>$M_1^*$ MJ($\hat{c}$)</td>
<td>0.46</td>
<td>0.23</td>
<td>0.28</td>
<td>0.02</td>
</tr>
<tr>
<td>$M_1^*$ MJD($\hat{c}$)</td>
<td>0.30</td>
<td>0.15</td>
<td>0.38</td>
<td>0.10</td>
</tr>
<tr>
<td>$M_1^*$ MJD(0)</td>
<td>0.42</td>
<td>0.23</td>
<td>0.32</td>
<td>0.30</td>
</tr>
<tr>
<td>$M_2^*$ SJ</td>
<td>0.02</td>
<td>0.07</td>
<td>0.18</td>
<td>0.04</td>
</tr>
<tr>
<td>$M_2^*$ FR</td>
<td>0.08</td>
<td>0.07</td>
<td>0.69</td>
<td>0.13</td>
</tr>
<tr>
<td>$M_2^*$ MJ($\hat{c}$)</td>
<td>0.23</td>
<td>0.24</td>
<td>0.52</td>
<td>0.00</td>
</tr>
<tr>
<td>$M_2^*$ MJD($\hat{c}$)</td>
<td>0.14</td>
<td>0.20</td>
<td>0.60</td>
<td>0.02</td>
</tr>
<tr>
<td>$M_2^*$ MJD(0)</td>
<td>0.23</td>
<td>0.25</td>
<td>0.52</td>
<td>0.00</td>
</tr>
<tr>
<td>$M_3^*$ SJ</td>
<td>0.03</td>
<td>0.06</td>
<td>0.39</td>
<td>0.05</td>
</tr>
<tr>
<td>$M_3^*$ FR</td>
<td>0.00</td>
<td>0.10</td>
<td>0.71</td>
<td>0.12</td>
</tr>
<tr>
<td>$M_3^*$ MJ($\hat{c}$)</td>
<td>0.15</td>
<td>0.14</td>
<td>0.71</td>
<td>0.00</td>
</tr>
<tr>
<td>$M_3^*$ MJD($\hat{c}$)</td>
<td>0.10</td>
<td>0.15</td>
<td>0.73</td>
<td>0.01</td>
</tr>
<tr>
<td>$M_3^*$ MJD(0)</td>
<td>0.14</td>
<td>0.15</td>
<td>0.71</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.04.
Standard error of Overall Bias ranges from 0.07 to 0.21.
Ratio of Standard error of Root MSE to Root MSE ranges from 0.14 to 0.27
3.2 Simulation 2: In the Presence of Noise Features

The robustness of clustering algorithms to dimensions with little information is a desirable property. The main purpose of this simulation study is to ascertain how robust these five methods tend to be in the presence of noise features/dimensions.

3.2.1 Simulation 2 Design

The simulation design is similar to Simulation 1 above with the following notable changes for dimension $p$, and the matrix of cluster centers, $M^*$. The data were generated to possess at least one “important” dimension:

1. The dimension of $X$, $p$.

Three levels of dimension, $D_1$, $D_2$ and $D_3$ were considered as follows:

$D_1$: The number of dimensions, $p$ is randomly generated from the set, \{2, 3, 4, \ldots, 10\}. This consists of “important” and noise dimensions. Let $p_I$ denote the number of “important” dimensions. Given $p$, we select $p_I$ randomly from the set, \{1, 2, 3, \ldots, p\}.

$D_2$: Here, the number of dimensions, $p$ is randomly generated from the set, \{11, 12, 13, \ldots, 50\}. The number of “important” dimensions, $p_I$ were generated the same way as in $D_1$.

$D_3$: The number of dimensions, $p$ is randomly generated from the set, \{51, 52, 53, \ldots, 100\}. The number of “important” dimensions, $p_I$ are generated in the same way as in $D_1$ and $D_2$.

2. The cluster centers, $M^*$. 

54
Three levels of cluster centers were considered. The matrix of cluster centers of each level $M_a^*$ ($a = 1, 2, 3$) were generated in two stages as follows:

First, for a fixed cluster number $g_0$ and a fix dimension $p$, let $M_a$ be a $g_0 \times p$ matrix. We construct $M_a$ to have the natural partition:

$$M_a = \begin{pmatrix} M_{1a} & M_{2a} \end{pmatrix},$$

where $M_{1a}$ is a $g_0 \times p_I$ matrix representing the centers of the “important” dimensions and $M_{2a}$ is a $g_0 \times (p - p_I)$ matrix representing the centers of noise dimensions. The elements of the matrix $M_{1a}$ are then generated randomly from the $N(0, a^2)$ distribution. The elements of $M_{2a}$ are fixed at 0, i.e., $M_{2a} = 0$, where $0$ is a $g_0 \times (p - p_I)$ matrix of zeros.

Secondly, we randomly permute the columns of $M_a$ to obtain our desired matrix of cluster centers, $M_a^*$ given by $M_a^* = (c_{a1}^T, \ldots, c_{a_{g_0}}^T)$, where $c_{aj}^T$, $j = 1, \ldots, g_0$ the $j$th row of $M_a^*$ represents the $p$-dimensional center of the $j$th cluster.

A $3 \times 3$ design based on the three levels of dimension, namely, $D_1$, $D_2$ and $D_3$ representing “small” dimensions, “medium” dimensions and “large” dimensions respectively and the three levels of cluster center separation $M_1^*$, $M_2^*$ and $M_3^*$, representing “small” separation, “medium” separation and “large” separation respectively, were employed in generating our simulated data sets. A total of 200 (consisting of 40 “blocks” each replicated five times) data sets are simulated from each of the nine treatment combination of the $3 \times 3$ factorial design.
Table 3.2: Frequency of Prediction Error For the Five methods: Small Cluster Center Separations $M_1^*$ and $p_1 \sim DU[\{1, \ldots, p\}]$

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Overall Bias($\hat{g}$)</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$ ($p \sim DU[{2, \ldots, 10}]$)</td>
<td>$SJ$</td>
<td>0.15 0.04 0.05 0.04 0.72</td>
<td>3.09 4.70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.52 0.31 0.12 0.04 0.01</td>
<td>-1.66 1.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.48 0.27 0.23 0.01 0.01</td>
<td>-1.46 1.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.46 0.28 0.23 0.02 0.01</td>
<td>-1.37 1.42</td>
<td></td>
</tr>
<tr>
<td>$D_2$ ($p \sim DU[{11, \ldots, 50}]$)</td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>5.63 5.69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.29 0.13 0.43 0.06 0.09</td>
<td>-0.67 1.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.30 0.38 0.31 0.00 0.01</td>
<td>-1.12 0.81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.28 0.31 0.40 0.01 0.00</td>
<td>-1.00 0.82</td>
<td></td>
</tr>
<tr>
<td>$D_3$ ($p \sim DU[{51, \ldots, 100}]$)</td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>5.45 5.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.43 0.23 0.16 0.03 0.00</td>
<td>-1.00 1.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.32 0.15 0.53 0.00 0.00</td>
<td>-0.97 0.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.25 0.12 0.62 0.00 0.01</td>
<td>-0.73 0.72</td>
<td></td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Overall Bias ranges from 0.10 to 0.25.
Ratio of Standard error of Root MSE to Root MSE ranges from 0.17 to 0.29

3.2.2 Simulation 2 Results

From Tables 3.2, 3.3 and 3.4, it is clear that the $SJ$ method cannot handle data with noise features. Overall, $MJD(\hat{c})$ appears to be the most robust method when noise features are present in data. The performance of $FR$ in general appears to deteriorate as the number of noise features in the data increases. The above assertion will be further investigated in our next simulation study. For small dimension cases $D_1$, $FR$ performs very poorly when the separation between cluster centers of the “important” dimensions is small, $M_1^*$. The performance of $FR$ however appears to improve as the the separation between cluster centers grows. In general $MJD(\hat{c})$ appears to be the best performer overall for Tables 3.2-3.4.
Table 3.3: Frequency of Prediction Error For the Five methods: Medium Cluster Center Separations $M_2^*$ and $p_I \sim DU[\{1, \ldots, p\}]

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Overall Bias($\hat{g}$)</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$ ($p \sim DU[[2, \ldots, 10]]$)</td>
<td>SJ</td>
<td>0.05 0.05 0.08 0.01 0.81</td>
<td>3.80</td>
<td>4.37</td>
</tr>
<tr>
<td></td>
<td>FR</td>
<td>0.17 0.20 0.43 0.16 0.04</td>
<td>-0.40</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>MJ($\hat{c}$)</td>
<td>0.36 0.32 0.31 0.00 0.01</td>
<td>-1.16</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>MJD($\hat{c}$)</td>
<td>0.27 0.30 0.34 0.05 0.04</td>
<td>-0.82</td>
<td>1.40</td>
</tr>
<tr>
<td></td>
<td>MJD(0)</td>
<td>0.34 0.33 0.32 0.01 0.00</td>
<td>-1.12</td>
<td>1.48</td>
</tr>
<tr>
<td>$D_2$ ($p \sim DU[[11, \ldots, 50]]$)</td>
<td>SJ</td>
<td>0.00 0.00 0.01 0.00 0.99</td>
<td>5.74</td>
<td>5.85</td>
</tr>
<tr>
<td></td>
<td>FR</td>
<td>0.03 0.10 0.62 0.16 0.09</td>
<td>0.22</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>MJ($\hat{c}$)</td>
<td>0.16 0.14 0.70 0.00 0.00</td>
<td>-0.49</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>MJD($\hat{c}$)</td>
<td>0.12 0.10 0.77 0.01 0.01</td>
<td>-0.36</td>
<td>0.84</td>
</tr>
<tr>
<td></td>
<td>MJD(0)</td>
<td>0.18 0.11 0.70 0.01 0.00</td>
<td>-0.52</td>
<td>1.00</td>
</tr>
<tr>
<td>$D_3$ ($p \sim DU[[51, \ldots, 100]]$)</td>
<td>SJ</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>5.25</td>
<td>5.37</td>
</tr>
<tr>
<td></td>
<td>FR</td>
<td>0.23 0.07 0.68 0.01 0.01</td>
<td>-0.86</td>
<td>1.69</td>
</tr>
<tr>
<td></td>
<td>MJ($\hat{c}$)</td>
<td>0.11 0.20 0.69 0.00 0.00</td>
<td>-0.45</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>MJD($\hat{c}$)</td>
<td>0.06 0.12 0.82 0.00 0.00</td>
<td>-0.26</td>
<td>0.65</td>
</tr>
<tr>
<td></td>
<td>MJD(0)</td>
<td>0.18 0.21 0.61 0.00 0.00</td>
<td>-0.63</td>
<td>1.09</td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Overall Bias ranges from 0.06 to 0.20.
Ratio of Standard error of Root MSE to Root MSE ranges from 0.17 to 0.30.
Table 3.4: Frequency of Prediction Error For the Five methods: Large Cluster Center Separations $M_3^*$ and $p_l \sim DU[\{1,\ldots,p\}]$

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Overall Bias($\hat{g}$)</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>SJ</td>
<td>$\leq -2$ $0.09$ $0.07$ $0.05$ $0.01$ $0.78$</td>
<td>3.69</td>
<td>4.70</td>
</tr>
<tr>
<td>$(p \sim DU[{2,\ldots,10}])$</td>
<td>FR</td>
<td>$0.21$ $0.20$ $0.44$ $0.09$ $0.06$</td>
<td>-0.56</td>
<td>1.42</td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>$0.32$ $0.24$ $0.42$ $0.02$ $0.00$</td>
<td>-1.04</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td>$MJ(D,\hat{c})$</td>
<td>$0.23$ $0.23$ $0.46$ $0.00$ $0.02$</td>
<td>-0.67</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>$MJ(D,0)$</td>
<td>$0.29$ $0.24$ $0.43$ $0.04$ $0.00$</td>
<td>-0.93</td>
<td>1.42</td>
</tr>
<tr>
<td>$D_2$</td>
<td>SJ</td>
<td>$0.00$ $0.00$ $0.02$ $0.00$ $0.98$</td>
<td>5.54</td>
<td>5.69</td>
</tr>
<tr>
<td>$(p \sim DU[{11,\ldots,50}])$</td>
<td>FR</td>
<td>$0.02$ $0.03$ $0.75$ $0.14$ $0.06$</td>
<td>0.21</td>
<td>1.09</td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>$0.06$ $0.20$ $0.74$ $0.00$ $0.00$</td>
<td>-0.36</td>
<td>0.81</td>
</tr>
<tr>
<td></td>
<td>$MJ(D,\hat{c})$</td>
<td>$0.04$ $0.10$ $0.86$ $0.00$ $0.00$</td>
<td>-0.21</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>$MJ(D,0)$</td>
<td>$0.06$ $0.20$ $0.74$ $0.00$ $0.00$</td>
<td>-0.37</td>
<td>0.82</td>
</tr>
<tr>
<td>$D_3$</td>
<td>SJ</td>
<td>$0.00$ $0.00$ $0.00$ $0.00$ $1.00$</td>
<td>5.57</td>
<td>5.67</td>
</tr>
<tr>
<td>$(p \sim DU[{51,\ldots,100}])$</td>
<td>FR</td>
<td>$0.12$ $0.09$ $0.78$ $0.00$ $0.01$</td>
<td>-0.51</td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>$0.10$ $0.06$ $0.84$ $0.00$ $0.00$</td>
<td>-0.31</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>$MJ(D,\hat{c})$</td>
<td>$0.09$ $0.05$ $0.86$ $0.00$ $0.00$</td>
<td>-0.26</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>$MJ(D,0)$</td>
<td>$0.10$ $0.07$ $0.83$ $0.00$ $0.00$</td>
<td>-0.32</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Overall Bias ranges from 0.06 to 0.26.
Ratio of Standard error of Root MSE to Root MSE ranges from 0.18 to 0.31
3.3 Simulation 3: Controlling Noise Features

In Simulation 2, it was difficult to directly ascertain the effect of the presence of noise features on the ability to estimate \( g_0 \) because the number of noise features was not controlled. In this simulation we control the proportion of noise features present at three levels. With the exception of the changes below, the rest of the design is the same as Simulation 2.

3.3.1 Simulation 3 Design

1. The dimension \( X, p \).

The dimension of the data \( p \) was randomly generated from the discrete set \( \{10, 12, 13, \ldots, 100\} \). Out of the \( p \) dimensions, the number of “important” dimensions was given by \( p_I = \alpha_p p \), where \( 0 < \alpha_p < 1 \). We set value of \( \alpha_p \) at three levels 0.25, 0.50 and 0.75. Integer values of \( p_I \) were used.

2. The cluster centers, \( M^* \).

All cluster centers were generated using the \( M_1^* \) design described in detail in Simulation 2.

3.3.2 Simulation 3 Results

We present the results of the above simulations in Table 3.5. As expected, the performance of all five methods tends to decrease with an increase in the proportion of noise features, \( 1-\alpha_p \). However, it is clear that our proposed methods tend to be more
Table 3.5: Frequency of Prediction Error For the Five methods: Small Cluster Center Separation $M_1^*$ and $p \sim DU[\{10, \ldots, 100\}]$

<table>
<thead>
<tr>
<th>Prop. of Important Features ($\alpha_p$)</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Overall Bias($\hat{\theta}$)</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\leq -2$</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$\alpha_p = 0.75$</td>
<td>$S$</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>$F$</td>
<td>0.15</td>
<td>0.17</td>
<td>0.34</td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.19</td>
<td>0.21</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>$MJD(\bar{c})$</td>
<td>0.08</td>
<td>0.13</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.20</td>
<td>0.11</td>
<td>0.68</td>
</tr>
<tr>
<td>$\alpha_p = 0.50$</td>
<td>$S$</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>$F$</td>
<td>0.33</td>
<td>0.25</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.21</td>
<td>0.32</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>$MJD(\bar{c})$</td>
<td>0.16</td>
<td>0.12</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.03</td>
<td>0.21</td>
<td>0.58</td>
</tr>
<tr>
<td>$\alpha_p = 0.25$</td>
<td>$S$</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>$F$</td>
<td>0.67</td>
<td>0.23</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.48</td>
<td>0.23</td>
<td>0.30</td>
</tr>
<tr>
<td></td>
<td>$MJD(\bar{c})$</td>
<td>0.36</td>
<td>0.20</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.45</td>
<td>0.20</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.04.
Standard error of Overall Bias ranges from 0.06 to 0.13
Ratio of Standard error of Root MSE to Root MSE ranges from 0.15 to 0.23

robust to the presence of noise dimensions than $F$.

### 3.4 Simulation 4: Based on Covariance Structure

For Simulation 4, in addition to the purpose stated in Simulation 2, we are particularly interested in how the covariance structure affects the clustering ability of each of the five methods.

#### 3.4.1 Simulation 4 Design

A $2 \times 2$ factorial arrangement based on the covariance structure was employed to obtained our simulated data. The covariance matrix of the $j$th cluster is given by
Let the matrices $V_j$ and $R_j$ as defined by (3.4) and (3.3) respectively be the two factors. $R_j$ is fixed at two levels, $\rho_j = 0$ and $\rho_j = 0.8$. We also set the factor $V_j$ at two levels. The first level ($A_1$) is specified by letting $\sigma_{ji} = 1$, $i = 1, \ldots, p$ in $V_j$ and the second level ($A_2$), $\sigma_{ji} = 2i/p + 1$, $i = 1, \ldots, p$ in $V_j$. For a fixed $p$, we note that both levels of $V_j$ have the mean of their diagonal elements being one and but their variances being zero and $p/3(p+1)$ respectively.

A total of 200 (consisting of 40 “blocks” each replicated five times) data sets are simulated from each of the four treatment combination of the $2 \times 2$ factorial design. All data sets were generated using the $M_1^*$ cluster center design of Simulation 2.

Except for the dimension of data set $p$, all other factors are generated using the design outlined in Simulation 1. Here, the dimension, $p$ was generated from the discrete set $\{2, 3, \ldots, 100\}$. Out of the $p$ dimensions, the number of “important” dimensions was randomly generated from the discrete set $\{1, \ldots, p\}$.

3.4.2 Simulation 4 Results

Note that for the $\rho_j = 0$ in $R_j$ cases (data sets with AR(1) correlational structure with parameter zero), all three of our proposed method appear to perform better than $SJ$ and $FR$. However, for the $\rho_j = 0.8$ in $R_j$ cases, $FR$ tends to outperform all the other methods but $MJD(\hat{c})$. This may mean that our proposed method $MJD(\hat{c})$ tends to be robust to the presence of moderate correlations among dimensions. It is also worth noting that as the dimension $p$ increases, the inter-correlations between several dimensions tend to weaken, i.e., approach zero. This is by virtue of the fact that all
Table 3.6: Frequency of Prediction Error For the Five methods: Small Cluster Center Separations $M_1^*, p \sim DU[\{2, \ldots, 100\}]$ and $p_1 \sim DU[\{1, \ldots, p\}]$

<table>
<thead>
<tr>
<th>Trt. Group</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Overall Bias($\hat{g}$)</th>
<th>Root MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_j = 0, A1$</td>
<td>$SJ$</td>
<td>0.01 0.02 0.09 0.00 0.88</td>
<td>4.67 5.17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.39 0.28 0.21 0.07 0.05</td>
<td>-1.17 2.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>0.34 0.21 0.44 0.00 0.01</td>
<td>-0.95 1.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.21 0.22 0.55 0.01 0.01</td>
<td>-0.67 1.37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.39 0.16 0.45 0.00 0.00</td>
<td>-1.14 1.65</td>
<td></td>
</tr>
<tr>
<td>$\rho_j = 0.8, A1$</td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>5.58 5.69</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.25 0.14 0.51 0.09 0.01</td>
<td>-0.86 1.73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>0.36 0.19 0.45 0.00 0.00</td>
<td>-1.08 1.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.25 0.17 0.56 0.01 0.01</td>
<td>-0.72 1.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.36 0.17 0.47 0.00 0.00</td>
<td>-1.02 1.51</td>
<td></td>
</tr>
<tr>
<td>$\rho_j = 0, A2$</td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>5.24 5.54</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.23 0.10 0.35 0.20 0.12</td>
<td>-0.84 2.18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>0.32 0.27 0.40 0.00 0.01</td>
<td>-1.10 1.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.29 0.20 0.47 0.01 0.03</td>
<td>-0.81 1.64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.32 0.26 0.41 0.00 0.01</td>
<td>-1.19 1.66</td>
<td></td>
</tr>
<tr>
<td>$\rho_j = 0.8, A2$</td>
<td>$SJ$</td>
<td>0.03 0.00 0.00 0.00 0.97</td>
<td>5.61 5.74</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$FR$</td>
<td>0.31 0.14 0.45 0.03 0.07</td>
<td>-0.24 1.81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJ(\hat{c})$</td>
<td>0.38 0.26 0.34 0.00 0.02</td>
<td>-1.05 1.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.30 0.19 0.47 0.03 0.01</td>
<td>-0.85 1.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(0)$</td>
<td>0.38 0.23 0.38 0.01 0.00</td>
<td>-1.03 1.52</td>
<td></td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.04.
Standard error of Overall Bias ranges from 0.08 to 0.15.
Ratio of Standard error of Root MSE to Root MSE ranges from 0.15 to 0.24.

Simulated data sets were generated to possess the $AR(1)$ covariance structure. This could also be the reason why our proposed methods appear to perform reasonable well even in the $\rho_j = 0.8$ in $R_j$ cases.
Chapter 4

Further Evaluations

4.1 Cluster Validation

The validation of clustering results is an extremely important issue in cluster analysis. By validation we mean testing the significance of the putative clusters. When a clustering method gets the number of clusters or groups right, the question still remains, is the clustering method clustering the right items? If an external criterion or “Gold Standard” is available, it would be easy to answer the above question by comparing the cluster solution or the partition induced by any clustering method to the “Gold Standard,” via some “measure of agreement.”

The lack of an external criterion in the above sense can make cluster validation a big challenge. Bock (1985) derived a statistical procedure for testing the clustering result of low-dimensional data. It is not clear how well it applies to high dimensional data. Also, Bayesian approaches to model-based clustering tend to assign confidence to
the resulting clusters via the Bayes Factor (Kass and Raftery, 1995). We should note, however, that the reliability of the Bayes Factors tends to deteriorate as the sample size decreases. Hence empirical evaluation should be the final judge of the usefulness of a clustering result.

More recent approaches to cluster validation are based on resampling (Monti et al., 2003 and Ben-Hur, Elisseeff and Guyon, 2002). These methods use different resampling schemes to simulate perturbations of the original data, in order to assess the stability of the clustering result with respect to sampling variability.

The underlying assumption of these methods is that the more stable the results are with respect to simulated perturbations, the more we can be confident that these clusters represent real structure. A major appeal of these methods is that, by using the data to simulate the perturbations, the resulting data tend to possess certain relevant dependencies among the dimensions of the data. Effective clustering algorithms can capitalize on these dependencies to determine the real cluster structure. A rigorous evaluation of the clustering results is, however, not possible as they do not explicitly model the assumptions underlying the data generating process.

4.2 Comparing Two Partitions

In this Section, we present some cluster validation metrics based on the assumption of the availability of an external criterion or a “Gold Standard.” These metrics are sometimes referred to as the “measures of agreement.” We shall refer to “Gold
Standard” partitions as *classes* and reserve the term *clusters* for partitions induced by a clustering method.

Given any two partitions of the same data set, a natural question arises: to what extend do the partitions agree or disagree? Suppose the state “1” corresponds to a pair of units in the same class or cluster in a partition and “0” otherwise. Let \( U \) and \( V \) be two partitions of the same data set, representing the *classes* and *clusters* respectively. The four possible patterns that could result by matching the partitions \( U \) and \( V \) (consisting of a string of 1’s and 0’s) is summarized by the contingency table below:

Table 4.1: The four possible patterns resulting from matching elements of two binary valued vectors

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( a_{11} )</td>
<td>( a_{10} )</td>
</tr>
<tr>
<td>0</td>
<td>( a_{01} )</td>
<td>( a_{00} )</td>
</tr>
</tbody>
</table>

We note from the above table that:

- \( a_{11} \) is the number of pairs of objects in the same class in \( U \) and in the same cluster in \( V \);
- \( a_{10} \) is the number of pairs of objects in same class in \( U \) but not in the same cluster in \( V \);
- \( a_{01} \) is the number of pairs of objects in same cluster in \( V \) but not in the same class in \( U \);
- \( a_{00} \) is the number of pairs of objects in a different class in \( U \) and in a different cluster in \( V \).
In another context, let $U$ and $V$ be picked at random so that the number of objects in the clusters and classes are fixed. Let $n_{ji}$ be the number of objects in cluster $j$ and also in class $i$. Let $n_{i.}$ and $n_{.j}$ be respectively the number of objects in class $i$ and cluster $j$. Table 4.2 is a contingency table for comparing $U$ to $V$.

Table 4.2: Contingency table for comparing two partitions.

| Classes | 1 | 2 | \cdots | u | Total |
|---------|---|---|\cdots|---|------|
| $1$     | $n_{11}$ | $n_{12}$ | \cdots | $n_{1u}$ | $n_1$. |
| $2$     | $n_{21}$ | $n_{22}$ | \cdots | $n_{2u}$ | $n_2$. |
| $\vdots$ | $\vdots$ | $\vdots$ | \cdots | $\vdots$ | $\vdots$ |
| $v$     | $n_{v1}$ | $n_{v2}$ | \cdots | $n_{vu}$ | $n_v$. |
| Total   | $n_{1.}$ | $n_{2.}$ | \cdots | $n_{u.}$ | $n$   |

It can be easily verified that

\[ a_{11} = \sum_{ij} \left( \frac{n_{ji}}{2} \right), \quad a_{10} = \sum_{i=1}^{u} \left( \frac{n_{i.}}{2} \right) - \sum_{ij} \left( \frac{n_{ji}}{2} \right), \]
\[ a_{01} = \sum_{j=1}^{v} \left( \frac{n_{j.}}{2} \right) - \sum_{ij} \left( \frac{n_{ji}}{2} \right), \quad a_{00} = \left( \frac{n}{2} \right) - a_{11} - a_{10} - a_{01}. \]

Several measures of agreement have been proposed in the above context. The abundance of these “measures of agreement” is causing confusion among investigators as to which measure to use in a study. Perhaps investigators should rely on the advice of Weisberg (1974, pp. 1652-1653):

“I would contend that analysts frequently should not seek a single measure and will never find a perfect measure. Different measures exist because there are different concepts to measure \cdots It is time to stop acting embarrassed about the supposed surplus of measures and instead make fullest possible use of their diversity.”

66
One of the most widely used measure of agreement is Rand’s measure of association also sometimes referred to as the Rand index (Rand 1971). This measure is based on the number of pairs of units which belong to the same cluster. The Rand index is given by
\[
\frac{a_{11} + a_{00}}{a_{11} + a_{10} + a_{01} + a_{00}}.
\]
This measure lies between 0 and 1. The Rand index is 1 when the two partitions \(U\) and \(V\) are identical. The expected value of this index, however does not take a constant value and this has been a major criticism of the Rand index.

Hubert and Arabie (1985) proposed the adjusted Rand index to correct for the above mentioned criticism of the Rand index. The adjusted Rand index is given by
\[
\frac{a_{11} - \vartheta}{(a_{10} + 2a_{11} + a_{01})/2 - \vartheta},
\]
where
\[
\vartheta = \frac{(a_{11} + a_{10})(a_{11} + a_{01})}{a_{11} + a_{10} + a_{01} + a_{00}}.
\]
This index is bounded above by 1 and has an expected value of zero in the case of random clusters. It assumes the generalized hypergeometric distribution as the model of randomness, i.e., the partitions \(U\) and \(V\) are picked at random but with the constraint that the number of objects in the classes and the number of clusters are fixed. For a detailed description of the adjusted Rand index see Yueng and Ruzzo (2001).

Another criticism of the Rand index is that it takes into account the joint absence of a feature by including \(a_{00}\) in its computation. Some authors such as Alexander
and Blasfield (1988) argue that if the Rand index is used, some cases may appear identical primarily because they both lack the same feature rather than because of the features they share in common. Even though this has not been an issue in most social sciences, the problem has arisen in archaeology. For example, if an object is absent from a burial, it's absence could be due either to cultural prescriptions or natural process of disintegration and attrition. It would therefore be inappropriate to base the estimation of the association between two burials on the joint absence of an artifact if it is impossible to know which of the two possible explanations is responsible for its absence. A measure of agreement that corrects for the above problem is Jaccard’s coefficient, defined as

\[
\frac{a_{11}}{a_{11} + a_{10} + a_{01}}.
\]

Jaccard’s coefficient avoids the use of the joint absence of a feature in its construction.

Another method for comparing two partitions was proposed by Fowlkes and Mallows (1983). Their method of agreement is defined as

\[
\frac{a_{11}}{\sqrt{(a_{11} + a_{10})(a_{11} + a_{01})}}. \tag{4.2}
\]

If \( U \) and \( V \) match perfectly, then (4.2) is one. If each group in \( U \) is equally distributed over all groups in \( V \), then (4.2) is zero.
4.3 A Comparative Study

4.3.1 Evaluation Methodology

Many factors such as the dimension $p$ of data, the presence of noise dimensions in the data, the total number of points $n$ in the data, cluster orientation, the underlying distribution of data, etc., tend to affect the overall performance of clustering methods. The importance of these factors may vary from method to method. Thus, in evaluating a clustering procedure, in addition to comparing it to existing methods it may be informative to identify the factors that tend to influence its performance most.

Our evaluation is aimed at:

- identifying the factors that tend to influence the cluster recovery ability of our proposed method $MJD(\hat{c})$;
- further comparing our proposed method $MJD(\hat{c})$ to the Model based approach of Fraley and Raftery (1998).

Below, we present some relevant summary statistics and evaluation metrics to be used for the above purposes.

4.3.1.1 Relevant Summary Statistics

In addition to factors like dimension $p$, the number of “important” dimensions $p_I$, the sample size $n$, and the number of clusters $g_0$ present in a data set, we define the following relevant summary statistics:
1. *MeanNg* and *SdNg*:

The mean and standard deviation of \( N_g = \{ (n_1, \ldots, n_g) : \sum_{j=1}^{g} n_j = n \} \), which are respectively *MeanNg* = \( n/g \) and *SdNg* = \( \sqrt{\frac{\sum_{j=1}^{g} (n_j - n/g)^2}{g}} \) provide useful summaries for our data analysis. They respectively seek to measure the mean number of points per cluster and the variability of the within cluster points. Most clustering methods tend to perform optimally when *SdNg* is close to zero.

2. *DetRho*:

This summary statistic is based on correlational structure present in data. It is well known that the higher the correlation among the variables the more difficult it is for most clustering methods to perform well. Let \( R_j \) be a \( p \times p \) correlation matrix associated with \( X_j \). Further suppose \( R_j \) has the AR(1) structure, that is

\[
R_j = \begin{pmatrix}
1 & \rho_j & \ldots & \rho_j^{p-1} \\
\rho_j & 1 & \ldots & \rho_j^{p-2} \\
\vdots & \vdots & \ddots & \vdots \\
\rho_j^{p-1} & \rho_j^{p-2} & \ldots & 1
\end{pmatrix}.
\]

It can easily be verified that the determinant of \( R_j \) is given by

\[
det(R_j) = (1 - \rho_j^2)^{p-1}.
\]

To measure the difficulty in retrieving clusters embedded in data based only on the above correlational structure, we propose

\[
DetRho = \frac{\sum_{j=1}^{g} (1 - \rho_j^2)^{p-1}}{g}.
\]
We note the following properties of $\text{DetRho}$:

- $0 \leq \text{DetRho} \leq 1$
- $\text{DetRho} = 0 \Leftrightarrow \rho_j = 1 \ \forall j$ (perfect correlation)
- $\text{DetRho} = 1 \Leftrightarrow \rho_j = 0 \ \forall j$ (no correlation)
- Given that $|\rho_j| < 1$, $\text{DetRho}$ approaches zero if and only if $p$ approaches infinity.

In general smaller values (close to zero) of $\text{DetRho}$ tend to imply that the $\rho_j$’s (which are not necessarily the same) are larger. Therefore it becomes difficult for most clustering methods to perform well when $\text{DetRho}$ is small. Similarly, large values of $\text{DetRho}$ indicates smaller $\rho_j$ values.

3. $SdSigma$:

This summary statistic is based on the variability of the $p$ variables. Let $V_j \ (j = 1, \ldots, g_0)$ be a $p \times p$ diagonal matrix defined by

$$V_j = \begin{pmatrix}
\sigma_{j1} & 0 & \ldots & 0 \\
0 & \sigma_{j2} & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots \\
0 & 0 & \ldots & \sigma_{jp}
\end{pmatrix},$$

where $\sigma_{ji}, \ i = 1 \ldots, p$ is the standard deviation of the $i$th variate within the $j$th cluster. The standard deviation of the diagonal elements of the above matrix, denoted by $SdSigma$ provides yet another useful data summary. In general, most clustering methods tend to perform best when $SdSigma$ is close to zero.

4. $SEPINDEX$: 

71
This is a measure of cluster separation. It is intuitively reasonable to assume that the smaller the distance is between cluster centers, the more difficult it would be for any clustering procedure to determine the true number of centers or groups in a data set. We should note, however, that cluster separation may not be the only factor that characterizes the ability of clustering procedures to determine the right number of clusters. Thus, even though a data set may contain well separated clusters, other factors such as the underlying distribution of the data and number of points per cluster may retard the performance of a particular method. Nevertheless, the separation between cluster centers is one of the most important determinants of clustering ability of many methods.

For any two clusters $i$ and $j$ containing $n_i$ and $n_j$ points, respectively, let $c_i$ and $c_j$ be their centers. Further suppose cluster $i$ and cluster $j$ have covariance structures given by $\Sigma_i$ and $\Sigma_j$, respectively. A natural measure of cluster separation between the two clusters $i$ and $j$ is the square Mahalanobis distance

$$M_{ij}^2 = (c_i - c_j)^T \Sigma_{ij}^{-1} (c_i - c_j),$$

(4.4)

where $\Sigma_{ij} = \Sigma_i/n_i + \Sigma_j/n_j$. For a data set containing $g$ clusters, we propose an overall cluster separation index

$$M_\cdot = \frac{1}{p\binom{g}{2}} \sum_{i<j} M_{ij}, \quad 1 \leq i < j \leq g,$$

(4.5)

where $p$ is the dimension of the data set. $M_\cdot$ is subsequently referred to as $SEPINDEX$ in our data analysis. We note that for the special case where
\( n_i = n_j = n \) and \( \Sigma_i = \Sigma_j = \Sigma \), (4.4) becomes

\[
M^2_{ij} = \frac{n(c_i - c_j)^T \Sigma^{-1} (c_i - c_j)}{2},
\]

which is analogous to the chi-square non centrality parameter. Another special case of interest is when the covariance structure \( \Sigma_{ij} \) is the identity matrix. Here, (4.4) becomes

\[
M^2_{ij} = ||c_i - c_j||^2,
\]

where \( ||\cdot|| \) is the Euclidean distance and (4.5) is

\[
M_n = \frac{1}{p(g)} \sum_{i<j} ||c_i - c_j||, \quad 1 \leq i < j \leq g.
\]

### 4.3.2 Evaluation Metrics

We evaluate the quality of the cluster solutions predicted by each of the three methods by measuring the agreement of the partitions induced by these methods to the known “true” partitions. Several measures of agreement have been proposed. In this study, we use the adjusted Rand index (Huber and Arabie, 1985) and the Fowlkes and Mallows (1983) index. They both have values between 0 and 1, with 1 corresponding to perfect agreement. These measures have been described in detail in Section 4.2.

Let \( \hat{g} \) be the predicted number of clusters associated with an arbitrary clustering method. We define yet another measure of agreement based on the “true” number
of clusters $g_0$ as

$$AEPCN = \frac{g - g_0}{g_0}. \quad (4.9)$$

This measure provides us with a sense of how well each method is performing in terms of recovering the “true” number of clusters in the data set. It is indeed the absolute error in predicting the cluster number per the “true” cluster number. $AEPCN = 0$ indicates that the estimated cluster number equals the “true” cluster number $g_0$. We should note that while the metrics $AdRand$ and $FM$ focus primarily on cluster membership, $AEPCN$ focuses mainly on the predictive ability of the method with respect to selecting the correct number of clusters. Table 4.3 provides a summary of some relevant abbreviations used throughout this study.

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$AdRand$</td>
<td>Adjusted Rand index</td>
</tr>
<tr>
<td>$FM$</td>
<td>Fowlkes and Mallows index</td>
</tr>
<tr>
<td>$AEPCN$</td>
<td>The absolute error in predicting</td>
</tr>
<tr>
<td></td>
<td>the cluster number defined by (4.9)</td>
</tr>
</tbody>
</table>

### 4.3.3 Artificial Data Set

For further evaluation purposes, we derive an artificial data set from Simulation 2, by summarizing each generated data in terms of its cluster composition such as the summary statistics given in Section 4.3.1.1. We shall refer to this data set as $Normalset$. The general description of this artificial data set is given in detail below.
4.3.3.1 Normalset

The purpose of this artificial data set is to determine the factors which tend to affect the performance of our $MJD(\hat{c})$ method and also to compare $MJD(\hat{c})$ to the Model based approach of Fraley and Raftery (1998). The Normalset data consist of 360 sample points and 18 variables. The variables and their descriptions are provided in Table 4.4.

Table 4.4: Description of variables in the Normalset data

<table>
<thead>
<tr>
<th>Variable</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$n$</td>
<td>Total number of points</td>
</tr>
<tr>
<td>2</td>
<td>$p$</td>
<td>Dimension of data</td>
</tr>
<tr>
<td>3</td>
<td>$p_1$</td>
<td>Number of important dimensions</td>
</tr>
<tr>
<td>4</td>
<td>$g$</td>
<td>Cluster number</td>
</tr>
<tr>
<td>5</td>
<td>$SEPINDEX$</td>
<td>Cluster separation index as defined by (4.8)</td>
</tr>
<tr>
<td>6</td>
<td>$SdN_g$</td>
<td>Standard deviation of $N_g$, the set of points within each cluster</td>
</tr>
<tr>
<td>7</td>
<td>$MeanN_g$</td>
<td>Mean of $N_g$</td>
</tr>
<tr>
<td>8</td>
<td>$DetRho$</td>
<td>This defined by (4.3)</td>
</tr>
<tr>
<td>9</td>
<td>$SdSigma$</td>
<td>The standard deviation of the diagonal elements of (3.4)</td>
</tr>
<tr>
<td>10</td>
<td>$Adrand_{SJ}$</td>
<td>The Adjusted rand index induced by $SJ$</td>
</tr>
<tr>
<td>11</td>
<td>$Adrand_{MJD(\hat{c})}$</td>
<td>The Adjusted rand index induced by the $MJD(\hat{c})$</td>
</tr>
<tr>
<td>12</td>
<td>$Adrand_{FR}$</td>
<td>The Adjusted rand index induced by $FR$</td>
</tr>
<tr>
<td>13</td>
<td>$FM_{SJ}$</td>
<td>The Fowlkes and Mallows index induced by $SJ$</td>
</tr>
<tr>
<td>14</td>
<td>$FM_{MJD(\hat{c})}$</td>
<td>The Fowlkes and Mallows index induced by $MJD(\hat{c})$</td>
</tr>
<tr>
<td>15</td>
<td>$FM_{FR}$</td>
<td>The Fowlkes and Mallows index induced by $FR$</td>
</tr>
<tr>
<td>16</td>
<td>$AEPCN_{SJ}$</td>
<td>The absolute predicting error induced by $SJ$</td>
</tr>
<tr>
<td>17</td>
<td>$AEPCN_{MJD(\hat{c})}$</td>
<td>The absolute predicting error induced by $MJD(\hat{c})$</td>
</tr>
<tr>
<td>18</td>
<td>$AEPCN_{FR}$</td>
<td>The absolute predicting error induced by $FR$</td>
</tr>
</tbody>
</table>
4.3.4 Factors Affecting the Performance of Method $MJD(\hat{c})$

It is well known that virtually every clustering method is based either implicitly or explicitly on several assumptions about the data generating process. The performance of these methods are often partly tied to how realistic these assumptions are with respect to the data to be clustered. The first nine variables listed in Table 4.4 are some of the factors well known to affect the performances of clustering methods.

We study how these variables tend influence the performance of our proposed method, $MJD(\hat{c})$. The performance of our $MJD(\hat{c})$ method with respect to how well it tends to predict cluster membership is measured by $AdRand_{MJD(\hat{c})}$, the Adjusted Rand index induced by $MJD(\hat{c})$, and $FM_{MJD(\hat{c})}$, the Fowlkes and Mallows index induced by $MJD(\hat{c})$. We also measure performance of $MJD(\hat{c})$ with respect to predicting the right cluster number using $AEPCN_{MJD(\hat{c})}$, the absolute error in predicting the cluster number.

Treating each evaluation metric ($AdRand_{MJD(\hat{c})}$, $FM_{MJD(\hat{c})}$ and $AEPCN_{MJD(\hat{c})}$) as a response to the first nine variables listed in Table 4.4, we analyze the Normalset data via PROC GLM in SAS.

We present the results in Table 4.5. The most important factors appear to be the cluster separation index $SEPINDEX$ and the number of important dimensions $p_I$ followed by the number of clusters $g$ present in the data and the number of dimension $p$.

The number of clusters $g_0$ in the data set appears to affect the performance
Figure 4.1: Performance of the $MJD(\tilde{c})$ method: The smooth curves are lowess fits (with $f=0.5$). Top panel: left: Adjusted Rand index versus Actual number of clusters. center: Fowlkes-Mallows index versus Actual number of clusters. right: $AEPCN$ versus Actual number of clusters. Middle panel: left: Adjusted Rand index versus Dimension. center: Fowlkes-Mallows index versus Dimension. right: $AEPCN$ versus Dimension. Bottom panel: left: Adjusted Rand index versus Important dimensions. center: Fowlkes-Mallows index versus Important dimensions. right: $AEPCN$ versus Important dimensions.
Figure 4.2: Performance of the MJD(\(\hat{c}\)) method: The smooth curves are lowess fits (with f=0.5) Top panel: left: Adjusted Rand index versus DetRho. center: Fowlkes-Mallows index versus DetRho. right: AEPCN versus DetRho. Middle panel: left: Adjusted Rand index versus SdSigma. center: Fowlkes-Mallows index versus SdSigma. right: AEPCN versus SdSigma. Bottom panel: left: Adjusted Rand index versus SEPINDEX. center: Fowlkes-Mallows index versus SEPINDEX. right: AEPCN versus SEPINDEX.
Table 4.5: ANOVA for Normalset data

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>$P$-value $\bar{N}_{MJD(\hat{c})}$</th>
<th>$P$-value $AdRand_{MJD(\hat{c})}$</th>
<th>$P$-value $FM_{MJD(\hat{c})}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>1</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>$g$</td>
<td>3</td>
<td>0.0001</td>
<td>0.10</td>
<td>0.14</td>
</tr>
<tr>
<td>$p$</td>
<td>1</td>
<td>0.003</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>$p_I$</td>
<td>1</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>$n$</td>
<td>1</td>
<td>0.68</td>
<td>0.58</td>
<td>0.61</td>
</tr>
<tr>
<td>DetRho</td>
<td>1</td>
<td>0.04</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>SdNq</td>
<td>1</td>
<td>0.35</td>
<td>0.50</td>
<td>0.41</td>
</tr>
<tr>
<td>SdSigma</td>
<td>1</td>
<td>0.26</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>MeanNq</td>
<td>1</td>
<td>0.51</td>
<td>0.46</td>
<td>0.48</td>
</tr>
<tr>
<td>SEPINDEX</td>
<td>1</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>

of $MJD(\hat{c})$ in terms of cluster number recovery but not in terms of cluster membership. Also $SdSigma$ appears to affect the performance of $MJD(\hat{c})$ in terms of cluster membership but not in terms of obtaining the cluster number recovery.

By the top panel of Figure 4.1, the performance (in terms of both cluster membership and fitting right cluster number) of our $MJD(\hat{c})$ method appears to decrease as the number of clusters in the data set increases. This decrease is however only significant in terms of obtaining the right cluster number. Although not shown here, it is worth mentioning that a similar pattern was observed for both the $SJ$ and $FR$ methods. This observation may, however, be due to the very close nature of cluster centers of the the simulated data rather than an inherent property of these methods.

Clustering methods tend to underestimate the cluster number $g_0$ for data sets containing many clusters with cluster centers close to each other. A good clustering method should, however, be independent of the number of clusters in the data set.

Because the information available for clustering is increasing as the number of
“important” dimensions $p_j$ increases, so does the performance of our method. This fact is clearly depicted in bottom panel of Figure 4.1. The same is true generally for the dimension $p$. See middle panel of Figure 4.1.

There appears to be a quadratic relationship between the performance of our method and the variability of the diagonal elements of (3.4) as measured by $SdSigma$. As clearly depicted by the middle panel of Figure 4.2, the performance of $MJD(\bar{c})$ appears to increase with an increase in the $SdSigma$ until $SdSigma = 0.4$, beyond which the performance tend to decrease.

Intuitively, one will expect the performance of a clustering method to increase as the separation between cluster centers increases. From Figure 4.2 it appears that the performance of our $MJD(\bar{c})$ method tends to increase as the separation between clusters increases up to a point and then levels off. This could mean that there is a threshold cluster center separation beyond which the clusters induced by $MJD(\bar{c})$ tend to be stable.

### 4.3.5 Comparing Method $MJD(\bar{c})$ to Method $FR$

To further compare $MJD(\bar{c})$ to $FR$, we first define the following relative differences:

- $AdRand_{23} = AdRand_{MJD(\bar{c})} - AdRand_{FR}$;
- $FM_{23} = FM_{MJD(\bar{c})} - FM_{FR}$;
- $AEPCN_{23} = AEPCN_{FR} - AEPCN_{MJD(\bar{c})}$.

80
where $AdRand_{MJD(\hat{c})}$ and $AdRand_{FR}$ are respectively the adjusted Rand index of the partitions induced by the $MJD(\hat{c})$ and $FR$ methods. $FM_{MJD(\hat{c})}$, $FM_{FR}$, $AEPCN_{MJD(\hat{c})}$ and $AEPCN_{FR}$ are similarly defined. By the above definitions, a positive relative difference for Adjusted Rand index, Fowlkes and Mallows index and $AEPCN$ indicates the superiority of $MJD(\hat{c})$ over $FR$.

The first nine variables listed on Table 4.4 were considered as possible experimental factors or explanatory variables for the above defined differences. Hence each of these relative differences was treated as a response to the nine variables listed in Table 4.4. Table 4.6 provides univariate statistics for the response variables.

Table 4.6: Univariate statistics

<table>
<thead>
<tr>
<th>Artificial Dataset</th>
<th>Response Variables</th>
<th>$AEPCN_{32}$</th>
<th>$AdRand_{23}$</th>
<th>$FM_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalset</td>
<td>$n$</td>
<td>360</td>
<td>360</td>
<td>360</td>
</tr>
<tr>
<td>Mean</td>
<td>0.13</td>
<td>0.01</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Std Dev</td>
<td>0.31</td>
<td>0.16</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>-0.80</td>
<td>-0.32</td>
<td>-0.32</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>1.8</td>
<td>0.58</td>
<td>0.47</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.7: ANOVA for Normalset data: $MJD(\hat{c})$ versus $FR$

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>$AEPCN_{32}$ P-value</th>
<th>$AdRand_{23}$ P-value</th>
<th>$FM_{23}$ P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>1</td>
<td>0.21</td>
<td>0.80</td>
<td>0.57</td>
</tr>
<tr>
<td>$g$</td>
<td>3</td>
<td>0.79</td>
<td>0.08</td>
<td>0.10</td>
</tr>
<tr>
<td>$p$</td>
<td>1</td>
<td>0.001</td>
<td>&lt; 0.0001</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>$p_I$</td>
<td>1</td>
<td>0.02</td>
<td>&lt; 0.0001</td>
<td>0.004</td>
</tr>
<tr>
<td>$n$</td>
<td>1</td>
<td>0.19</td>
<td>0.66</td>
<td>0.69</td>
</tr>
<tr>
<td>DetRho</td>
<td>1</td>
<td>0.11</td>
<td>0.44</td>
<td>0.30</td>
</tr>
<tr>
<td>SdNg</td>
<td>1</td>
<td>0.07</td>
<td>0.60</td>
<td>0.38</td>
</tr>
<tr>
<td>SdSigma</td>
<td>1</td>
<td>0.22</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>MeanNg</td>
<td>1</td>
<td>0.04</td>
<td>0.30</td>
<td>0.31</td>
</tr>
<tr>
<td>SEPINDEX</td>
<td>1</td>
<td>0.02</td>
<td>0.99</td>
<td>0.50</td>
</tr>
</tbody>
</table>
Overall, as indicated by the results in Table 4.7 (p-values of intercept) there seems to be no difference in the performance rate of $MJD(\hat{c})$ and $FR$ at a significance level of 0.05. There is, however, significance difference in the performance between these methods in terms of the Fowlkes and Mallow index and $AEPCN$. Table 4.7 also suggest that besides dimension, $p$, the number of “important” dimensions $p_I$ and the cluster separation index, $SEPINDEX$, the other factors tend not to be useful predictors of the differences $AdRand_{23}$, $FM_{23}$, and $AEPCN_{32}$. 
Figure 4.3: $MJD(\hat{c})$ versus $FR$ : The smooth curves are lowess fits (with $f=0.5$) Top panel: \textbf{left:} Mean difference in Adjusted Rand index versus Dimension \textbf{center:} Mean difference in Fowlkes-Mallows index versus Dimension \textbf{right:} Mean difference in $AEPCN$ versus Dimension Middle panel: \textbf{left:} Mean difference in Adjusted Rand index versus Important dimensions \textbf{center:} Mean difference in Fowlkes-Mallows index versus Important dimensions \textbf{right:} Mean difference in $AEPCN$ versus Important dimensions Bottom panel: \textbf{left:} Difference in Adjusted Rand index versus $SEPINDEX$ \textbf{center:} Difference in Fowlkes-Mallows index versus $SEPINDEX$ \textbf{right:} Difference in $AEPCN$ versus $SEPINDEX$
Chapter 5

Clustering Data with a Common Covariance Structure

The k-means algorithm, which is based on the trace criterion (pooled within cluster sum of squares), tends to be optimal when the subpopulations in a data set are all normally distributed with unknown mean vectors and the same spherical (identity) covariance matrices (Gallegos and Ritter, 2005). It is also well known that the performance of the k-means algorithm tends to deteriorate as the correlation among dimensions/features increases.

In clustering applications where the non-spherical covariance structure within each group is known to be roughly the same, clustering results of the k-means algorithm could be improved by first transforming the data to possess the identity covariance matrix.

We formalize these ideas as follows. Suppose \( \mathcal{X} \) is an \( n \times p \) data set, where:
(i) \(X\) contains \(g_0\) groups, where \(g_0 \in \{1, \ldots, g_{\text{max}}\}\);

(ii) The covariance structure within each group is the same.

That is,

\[
\Sigma_1 = \Sigma_2 = \cdots = \Sigma_{g_0} = \Sigma.
\]

Let \(\Sigma\) be a \(p \times p\) matrix defined by

\[
\Sigma = VRV, \quad (5.1)
\]

where \(V\) and \(R\) are \(p \times p\) matrices defined by

\[
R = \begin{pmatrix}
1 & \rho & \ldots & \rho^{p-1} \\
\rho & 1 & \ldots & \rho^{p-2} \\
\vdots & \vdots & \ddots & \vdots \\
\rho^{p-1} & \rho^{p-2} & \ldots & 1
\end{pmatrix}
\]

and

\[
V = \begin{pmatrix}
\sigma_1 & 0 & \ldots & 0 \\
0 & \sigma_2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \sigma_p
\end{pmatrix}.
\]

Consider the following transformation

\[
X_{\text{trfm}} = X \Sigma^{-\frac{1}{2}}
\]

\[
= X \Sigma_{g_0}^{-\frac{1}{2}}.
\]

Note that the covariance structure of the transformed data \(X_{\text{trfm}}\) is the identity matrix. That is, each group within \(X_{\text{trfm}}\) has the identity covariance structure. Thus one expects the k-means algorithm to be more effective in clustering \(X_{\text{trfm}}\) than \(X\). However, \(\Sigma\) is typically unknown, thus making the transformation given by (5.3) impractical.

We propose the following procedure for estimating the number of groups in \(X\):

85
1. First, preprocess $\mathcal{X}$ by performing following transformation:

$$\tilde{\mathcal{X}}_{trfm} = \mathcal{X} \hat{\Sigma}_{g_{max}}^{-\frac{1}{2}}, \quad (5.4)$$

where $\hat{\Sigma}_{g_{max}}$ is the estimate of the common covariance matrix assuming a mixture of normals model and setting the number of components equal to the maximum candidate cluster number $g_{max}$. In practice, we use model “EEE” (same as Model 3 of Table 2.2) of the MCLUST software (Fraley and Raftery, 1998) to estimate $\Sigma_{g_{max}}$. This model assumes the same covariance structure for each group.

2. Apply the clustering method on $\tilde{\mathcal{X}}_{trfm}$ to estimate the number of groups in $\mathcal{X}$.

The approach whereby we apply $MJD(\hat{c})$ to the preprocessed data $\tilde{\mathcal{X}}_{trfm}$ in estimating the number of clusters in $\mathcal{X}$ shall be referred to simply as $MJD_{pp}(\hat{c})$. Similarly, $SJ_{pp}$ is the approach of applying $SJ$ to the preprocessed data $\tilde{\mathcal{X}}_{trfm}$ in estimating the number of groups in $\mathcal{X}$.

5.0.6 Simulation

We study the preprocessing approach described above by comparing $MJD_{pp}(\hat{c})$ and $SJ_{pp}$ to $FR$, $MJD(\hat{c})$ and $SJ$ via a Monte Carlo simulation. We describe the simulation design below.
5.0.7 Simulation Design

We employ a $2 \times 2$ factorial arrangement based on the covariance structure (5.1) in generating our data. Let the $p \times p$ matrices $V$ and $R$ constitute our two factors. $R$ is fixed at two levels, $\rho = 0$ and $\rho = 0.8$. We also set the factor $V$ at two levels. The first level $A_1$ is specified by letting $\sigma_i = 1, \ i = 1, \ldots, p$ in $V$ and the second level $A_2, \ 
\sigma_i = \sqrt{2i/(p+1)}, \ i = 1, \ldots, p$ in $V$. The levels of the matrices $R$ and $V$ were chosen such that the trace of the common covariance matrix $\Sigma$ resulting from each of the four treatment combinations is $p$.

The $p$-dimensional cluster centers of all data sets were generated to consist of two parts: the “important” part, which we independently generated from $N(0, 2)$, and a noise part with the cluster centers all set to zero. In all, 20 unique cluster centers were generated from each of the 4 treatment groups. Each of these unique cluster centers was replicated 5 times. In addition, all data sets were generated to possess the following fixed attributes:

- The “true” cluster number $g_0$ and the total sample size $n$ were fixed at $g_0 = 4$ and $n = 150$ respectively;
- 3 levels of dimension $p$ were considered: $p = 10$, $p = 50$ and $p = 100$;
- For each level of dimension, we fix the number of “important” dimensions at $p_I = 2\sqrt{p}$. 

87
5.0.8 Simulation Results

Tables 5.1-5.4 present the results. Overall, method $MJD_{pp}(\hat{c})$ appears to be the best method for the large $p$ cases ($p = 50$ and $p = 100$). For the $p = 10$ cases, $MJD_{pp}(\hat{c})$ appears to be best method only for cases in which $\rho = 0.8$ in $\mathbb{R}$. However, as depicted in Tables 5.1 and 5.3, when $\rho = 0$ in $\mathbb{R}$, $FR$ tends to be the best method.

As expected, $MJD_{pp}(\hat{c})$ and $SJ_{pp}$ tend to be improvements over $MJD(\hat{c})$ and $SJ$ respectively, for cases in which the common covariance structure is non-spherical, i.e., $\Sigma \neq I_p$, the $p$-dimensional identity matrix. The results of these cases are shown in Tables 5.2, 5.3 and 5.4. Table 5.1 presents the cases in which the common covariance structure $\Sigma$ is the identity matrix. Under this condition, we expect the partitions induced by k-means to be optimal. Therefore preprocessing the data by (5.3) before clustering may not be helpful. This is reflected in Table 5.1 by the fact that the results of $SJ$ are not improved by $SJ_{pp}$. Also, note that the results of method $MJD(\hat{c})$ are slightly better than those of $MJD_{pp}(\hat{c})$.

We note that for the $p = 10$ cases, the results of $SJ_{pp}$ are a great improvement over those of $SJ$. This improvement is, however, small for the $p = 50$ and the $p = 100$ cases.

It is also worth noting that the performance of $FR$ tends to deteriorate as $p$ increases for the cases in which $\rho = 0$ in $\mathbb{R}$. This fact is contained in Tables 5.1 and 5.3. The performance of $FR$, however, tends to increase from $p = 10$ to $p = 50$ and then decreases to its lowest at $p = 100$ for the cases in which $\rho = 0.8$ in $\mathbb{R}$. See Tables 5.2 and
Figure 5.1: Performance of FR for cases in which $\rho = 0.8$ in $\mathbf{R}$ and $\sigma_i = 1, \ i = 1, \ldots, p$:
(a) Plot of mean adjusted Rand index versus dimension $p$; (b) Plot of mean AEPCN versus dimension $p$. 

89
5.4. In Figure 5.1, we depict the performance of $FR$ as $p$ increases for the cases in which $\rho = 0.8$ in $\mathbf{R}$ and $\sigma_i = 1, \ i = 1 \ldots, p$ in $\mathbf{V}$. Clearly, a threshold $p$ in the neighborhood of $p = 50$ exist beyond which the performance of $FR$ tends to deteriorate as $p$ becomes larger and larger. A possible explanation of the increase in performance from $p = 10$ to $p = 50$ is that the correlations between many features/dimensions tend to weaken (approach zero) as we move from $p = 10$ to $p = 50$. The decrease in performance of $FR$ from $p = 50$ to $p = 100$ appears to be due to the corresponding increase in the number of noise dimensions/features.

Table 5.1: Frequency of Prediction Error For the Five methods: $n = 150, p_I = 2\sqrt{p}$, $\rho = 0$ in $\mathbf{R}$ and $\sigma_i = 1, (i = 1, \ldots, p)$ in $\mathbf{V}$

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Mean $AEP$</th>
<th>Mean $CN$</th>
<th>Mean $AdRand$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p = 10$</td>
<td>$FR$</td>
<td>$0.00$ $0.02$ $0.98$ $0.00$ $0.00$</td>
<td>$0.01$ $0.95$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>$0.00$ $0.03$ $0.71$ $0.00$ $0.26$</td>
<td>$0.33$ $0.79$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>$0.00$ $0.01$ $0.38$ $0.61$ $0.00$</td>
<td>$0.67$ $0.72$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>$0.17$ $0.23$ $0.60$ $0.00$ $0.00$</td>
<td>$0.14$ $0.85$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>$0.09$ $0.31$ $0.59$ $0.00$ $0.01$</td>
<td>$0.14$ $0.86$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 50$</td>
<td>$FR$</td>
<td>$0.13$ $0.16$ $0.59$ $0.08$ $0.04$</td>
<td>$0.16$ $0.90$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>$0.00$ $0.00$ $0.00$ $0.00$ $1.00$</td>
<td>$1.25$ $0.65$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>$0.00$ $0.00$ $0.00$ $0.00$ $1.00$</td>
<td>$1.25$ $0.67$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>$0.00$ $0.15$ $0.85$ $0.00$ $0.00$</td>
<td>$0.04$ $0.91$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>$0.02$ $0.12$ $0.86$ $0.00$ $0.00$</td>
<td>$0.04$ $0.83$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 100$</td>
<td>$FR$</td>
<td>$0.89$ $0.03$ $0.01$ $0.02$ $0.05$</td>
<td>$0.71$ $0.49$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>$0.00$ $0.00$ $0.00$ $0.00$ $1.00$</td>
<td>$1.25$ $0.65$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>$0.00$ $0.00$ $0.00$ $0.00$ $1.00$</td>
<td>$1.24$ $0.68$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>$0.01$ $0.09$ $0.90$ $0.00$ $0.00$</td>
<td>$0.03$ $0.90$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>$0.01$ $0.05$ $0.84$ $0.00$ $0.10$</td>
<td>$0.14$ $0.85$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Mean $AEP$ $CN$ ranges from 0.00 to 0.10
Standard error of Mean $AdRand$ ranges from 0.01 to 0.04
Table 5.2: Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0.8$ in $\mathbf{R}$ and $\sigma_i = 1$, $(i = 1, \ldots, p)$ in $\mathbf{V}$

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Mean</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\leq -2$ &amp; $-1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p = 10$</td>
<td>$FR$</td>
<td>0.00 0.00 0.79 0.14 0.07</td>
<td>0.07 0.89</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.21 0.61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.65 0.00 0.35</td>
<td>0.35 0.81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.25 0.31 0.35 0.04 0.05</td>
<td>0.25 0.78</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.08 0.08 0.84 0.00 0.00</td>
<td>0.06 0.87</td>
<td></td>
</tr>
<tr>
<td>$p = 50$</td>
<td>$FR$</td>
<td>0.00 0.00 0.99 0.01 0.00</td>
<td>0.00 0.99</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.25 0.63</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.24 0.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.15 0.32 0.53 0.00 0.00</td>
<td>0.16 0.87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.00 0.04 0.96 0.00 0.00</td>
<td>0.01 0.95</td>
<td></td>
</tr>
<tr>
<td>$p = 100$</td>
<td>$FR$</td>
<td>0.35 0.15 0.19 0.05 0.26</td>
<td>0.51 0.72</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.25 0.62</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.24 0.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.08 0.15 0.70 0.00 0.00</td>
<td>0.08 0.83</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.02 0.13 0.85 0.00 0.00</td>
<td>0.04 0.92</td>
<td></td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Mean $AEPCN$ ranges from 0.00 to 0.10
Standard error of Mean $AdRand$ ranges from 0.01 to 0.04
Table 5.3: Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0$ in $\mathbb{R}$ and $\sigma_i = \sqrt{\frac{2i}{p+1}}$ ($i = 1, \ldots, p$) in $V$

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Mean $AEPCN$</th>
<th>Mean $AdRand$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p = 10$</td>
<td>$FR$</td>
<td>0.03 0.07 0.90 0.00 0.00</td>
<td>0.03 0.93</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.48 0.01 0.51</td>
<td>0.62 0.71</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.53 0.00 0.47</td>
<td>0.51 0.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.26 0.15 0.58 0.01 0.00</td>
<td>0.17 0.84</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.09 0.15 0.76 0.00 0.00</td>
<td>0.08 0.86</td>
<td></td>
</tr>
<tr>
<td>$p = 50$</td>
<td>$FR$</td>
<td>0.10 0.24 0.62 0.02 0.02</td>
<td>0.13 0.83</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.25 0.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.24 0.67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.00 0.12 0.88 0.00 0.00</td>
<td>0.03 0.87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.00 0.11 0.89 0.00 0.00</td>
<td>0.03 0.87</td>
<td></td>
</tr>
<tr>
<td>$p = 100$</td>
<td>$FR$</td>
<td>1.00 0.00 0.00 0.00 0.00</td>
<td>0.75 0.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.25 0.66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.24 0.68</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.00 0.09 0.91 0.00 0.00</td>
<td>0.02 0.88</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.01 0.02 0.97 0.00 0.00</td>
<td>0.01 0.94</td>
<td></td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Mean $AEPCN$ ranges from 0.00 to 0.10
Standard error of Mean $AdRand$ ranges from 0.01 to 0.04
Table 5.4: Frequency of Prediction Error For the Five methods: $n = 150$, $p_I = 2\sqrt{p}$, $\rho = 0.8$ in $\mathbf{R}$ and $\sigma_i = \sqrt{\frac{2}{p+1}}$, $(i = 1, \ldots, p)$ in $\mathbf{V}$

<table>
<thead>
<tr>
<th>Dimension, $p$</th>
<th>Method</th>
<th>Prediction Error</th>
<th>Mean $AEPCN$</th>
<th>Mean $AdRand$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p = 10$</td>
<td>$FR$</td>
<td>0.00 0.00 0.62 0.28 0.10</td>
<td>0.14 0.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.21 0.60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.73 0.01 0.26</td>
<td>0.26 0.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.23 0.35 0.23 0.06 0.13</td>
<td>0.32 0.77</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.02 0.05 0.93 0.00 0.00</td>
<td>0.02 0.89</td>
<td></td>
</tr>
<tr>
<td>$p = 50$</td>
<td>$FR$</td>
<td>0.00 0.03 0.79 0.15 0.03</td>
<td>0.06 0.90</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.25 0.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.23 0.68</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.41 0.39 0.18 0.02 0.00</td>
<td>0.31 0.82</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.01 0.03 0.96 0.00 0.00</td>
<td>0.01 0.95</td>
<td></td>
</tr>
<tr>
<td>$p = 100$</td>
<td>$FR$</td>
<td>0.46 0.24 0.19 0.05 0.06</td>
<td>0.40 0.73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.25 0.64</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$SJ_{pp}$</td>
<td>0.00 0.00 0.00 0.00 1.00</td>
<td>1.23 0.66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD(\hat{c})$</td>
<td>0.09 0.36 0.55 0.00 0.00</td>
<td>0.14 0.84</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$MJD_{pp}(\hat{c})$</td>
<td>0.03 0.11 0.86 0.00 0.00</td>
<td>0.04 0.89</td>
<td></td>
</tr>
</tbody>
</table>

Standard error of Prediction Error is bounded by 0.05.
Standard error of Mean $AEPCN$ ranges from 0.00 to 0.10
Standard error of Mean $AdRand$ ranges from 0.01 to 0.04
Chapter 6

Applications to Large $p$, Small $n$ Data Sets

Problems under the “large $p$, small $n$” paradigm are not only challenging conceptually but also computationally. This is particularly true in clustering. Most clustering algorithms tend to produce poor cluster solutions in such situations. Unfortunately, the “large $p$, small $n$” paradigm arises in many applications including microarray studies, where expression levels of thousands of genes are monitored for a small number of subjects. There is therefore the need to develop clustering techniques that can efficiently handle these kind of data.

Recently, Tibshirani and Walther (2005) proposed a criterion for determining the number of clusters based on a measure of prediction error of cluster membership called the prediction strength. They demonstrated the ability of their method to handle “large $p$, small $n$” data sets. The basic idea of their method is outlined as follows:
Given an $n \times p$ data set $\mathcal{X}$, the method first partitions it into two parts, a training set, $\mathcal{X}_{tr}$, and a test set, $\mathcal{X}_{te}$. Next, both the training set and test set are independently partitioned into $g$ polygonal regions. The centroids induced by clustering the training set into $g$ groups are then used to predict cluster membership in the test set. The method then estimates how well the centroids derived from the training set tend to predict cluster membership in the test set. The above ideas are formalized as follows:

Let $c(\mathcal{X}_{tr}, g)$ denote the clustering of the training set $\mathcal{X}_{tr}$ into $g$ regions. Let $D[c(\mathcal{X}_{tr}, g), \mathcal{X}_{te}]$ be an $n_{te} \times n_{te}$ matrix, with the $ii'$ element $D[c(\mathcal{X}_{tr}, g), \mathcal{X}_{te}]ii' = 1$ if observations $i$ and $i'$ of test set $\mathcal{X}_{te}$ fall into the same polygonal region induced by $c(\mathcal{X}_{tr}, g)$ and zero otherwise. For a candidate cluster number, $g \in \{1, \ldots, g_{max}\}$, let $I_j (j = 1, \ldots, g)$ be an index for observations in the test set that belong to the test cluster $j$, and let $n_j$ be the number of observations in the test cluster $j$. For $g = 1, \ldots, g_{max}$, we define the “prediction strength” by

$$ps(g) = \min_{1 \leq j \leq g} \frac{1}{n_j(n_j - 1)} \sum_{i' \in I_j} D[c(\mathcal{X}_{tr}, g), \mathcal{X}_{te}]ii'.$$

Note that $ps(1) = 1$, as both the training and test sets fall into just one cluster. In the absence of a test sample, repeated $r$-fold cross validation may be used to estimate $ps(g)$. Here, the $r$th fold is taken to be our test sample and the first $r - 1$ folds our training set. In general, for some $c_0 \in (0, 1]$, Tibshirani and Walther (2005) estimate
the cluster number by the largest $g$ such that

$$\text{ps}(g) + \text{se}(g) \geq c_0,$$

where $\text{se}(g)$ is the standard error of the prediction strength over the $r$ cross-validation fold. They suggest that $c_0$ equal to 0.8 or 0.9 tends to work well in practice.

We compare our $MJD(\hat{c})$ method to the prediction strength approach of Tibshirani and Walther (2005) via Monte Carlo simulations in the next Section. We shall sometimes refer to the prediction strength method of Tibshirani and Walther (2005) simply as $TW(c_0)$.

6.1 Simulations

To compare the clustering ability of our proposed method $MJD(\hat{c})$ to the prediction strength method of Tibshirani and Walther, (2005), 50 realizations of data were generated from each of the settings below. All simulated data sets were generated to mimic real microarray data.

**S1:** Two clusters in 6000 dimensions: Clusters 1 and 2 were generated to contain 6 standard normal observations each, with the first 1000 dimensions in Cluster 1 shifted by -2. That is, suppose $\mathcal{X}$ is an instance of the data generated here. Then $\mathcal{X}$ is a $12 \times 6000$ data set. For $i = 1, \ldots, 12$ and $j = 1, \ldots, 6000$, the data generating
model of the \((i,j)\)th element of \(X\) is given by

\[
x_{ij} = \mu_{ij} + Z_{ij}, \quad Z_{ij} \sim N(0,1),
\]

where \(\mu_{ij}\) is the mean structure of \(X_{ij}\), and is given by

\[
\mu_{ij} = \begin{cases} 
-2 & i = 1, 2, \ldots, 6, \quad j = 1, 2, \ldots, 100, \\
2 & i = 67, \ldots, 99, \quad j = 1, 2, \ldots, 100, \\
0 & \text{otherwise.}
\end{cases}
\]

**S2: Three normal clusters in 1000 dimensions (Tibshirani and Walter, 2005):**

Each of the three clusters were generated to posses 33 standard normal observations in 1000 dimensions, with the first 100 dimensions of each of the three clusters shifted by \(-2, 0\) and \(2\) respectively. That is, for \(i = 1, \ldots, 99\) and \(j = 1, \ldots, 100\) the data generating model of the \((i,j)\)th element of \(X\) is given by

\[
x_{ij} = \mu_{ij} + Z_{ij}, \quad Z_{ij} \sim N(0,1),
\]

where \(\mu_{ij}\) is the mean structure of \(x_{ij}\) and is given by

\[
\mu_{ij} = \begin{cases} 
-2 & i = 1, \ldots, 33, \quad j = 1, 2, \ldots, 100, \\
2 & i = 67, \ldots, 99, \quad j = 1, 2, \ldots, 100, \\
0 & \text{otherwise.}
\end{cases}
\]
**S3: Three lognormal clusters in 500 dimensions:**

We describe the data generating model as follows:

For \( i = 1, \ldots, 30 \) and \( j = 1, \ldots, 500 \) the data generating model of the \((i, j)\)th element of \( X \) is given by

\[
    x_{ij} = \exp \left[ 0.5 \left( \mu_{ij} + Z_{ij} \right) \right], \quad Z_{ij} \sim N(0, 1),
\]

where \( \mu_{ij} \) is given by

\[
    \mu_{ij} = \begin{cases} 
        -2 & i = 1, \ldots, 10, \quad j = 1, 2, \ldots, 140, \\
        2 & i = 21, \ldots, 30, \quad j = 1, 2, \ldots, 140, \\
        0 & \text{otherwise}. 
    \end{cases}
\]

**S4: Four normal clusters in 1000 dimensions:** The four clusters each containing 15 points have a common covariance structure the identity matrix. The mean vector of each of the four clusters were randomly generated from the standard normal distribution.

We present the results of the above simulations in Table 6.1. Our proposed method \( MJD(\hat{c}) \) appears to perform slightly better than the prediction strength method of Tibshirani and Walther (2005) with respect to detecting normal clusters. See S2 and S4. However, as indicated by the results of S3, the prediction strength method of Tibshirani and Walther (2005), appears to perform better than our \( MJD(\hat{c}) \) method for non-normal clusters.
Table 6.1: Simulation results: Large $p$, Small $n$ Situations

<table>
<thead>
<tr>
<th>Clu. No.</th>
<th>Simulation Method</th>
<th>Cluster Number Estimates</th>
<th>( g_0 - 2 )</th>
<th>( g_0 - 1 )</th>
<th>( g_0 )</th>
<th>( g_0 + 1 )</th>
<th>( g_0 + 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>S1</td>
<td>( TW(0.8) )</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( TW(0.9) )</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( MJD(\hat{c}) )</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>S2</td>
<td>( TW(0.8) )</td>
<td>0</td>
<td>2</td>
<td>48</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( TW(0.9) )</td>
<td>0</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( MJD(\hat{c}) )</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>S3</td>
<td>( TW(0.8) )</td>
<td>5</td>
<td>6</td>
<td>34</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( TW(0.9) )</td>
<td>0</td>
<td>15</td>
<td>30</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( MJD(\hat{c}) )</td>
<td>0</td>
<td>40</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>S4</td>
<td>( TW(0.8) )</td>
<td>0</td>
<td>4</td>
<td>46</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( TW(0.9) )</td>
<td>0</td>
<td>7</td>
<td>43</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( MJD(\hat{c}) )</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

6.2 Clustering Microarray Data Via The Maximal Jump Difference \( MJD(\hat{c}) \) Approach

DNA microarray data typically consist of the expression levels of thousands of genes that are often recorded over only a few tens of different samples. More often than not, the sizes of these data sets coupled with their biological complexity makes the use of multivariate clustering techniques invaluable in extracting correlated patterns and or natural classes that may be present.

Microarray data are often organized as gene expression matrices, with rows representing genes and columns environmental conditions or samples such as tissues. The entries of these matrices tend to represent the expression/activity of a particular gene under a given condition. Hence, it makes sense to cluster microarray data in two ways.

First, we may view the \( n_s \) samples as the \( n = n_s \) objects to be clustered, with
the \( n_g \) gene level expressions being features (dimensions) representing each sample as a point in a \( p = n_g \)-dimensional space. Here, we are in fact grouping samples according to their relatedness in expression patterns.

Real microarray data tend to be very noisy with no gene perfectly consistent within the different samples. This fact together with the fact that the number of dimensions \( n_g \), typically far exceeds the number of samples \( n_s \), makes the clustering of samples particularly challenging.

The other way is to regard the \( n_g \) genes as the \( n \) objects to be clustered, each representing its expression profile as a point in \( p = n_s \)-dimensional space. Clustering in this sense allows biologists to identify potentially meaningful relationships between genes by “guilt of association” or their mere appearance in the same cluster solution.

The two most commonly used clustering approaches in gene expression data analysis is hierarchical clustering (Eisen et al., 1988) and k-means (Tavazoie et al., 1999). Other approaches in the literature include self-organizing maps (Tamayo et al., 1999) and support vector machines (Brown et al., 2000).

Our proposed methods can easily handle problems where dimension \( p \) far exceeds the number of samples \( n \). They also tend to be robust to noise dimensions, and thus should prove useful in clustering microarray data.
6.2.1 Data Sets

To evaluate our proposed $MJD(\hat{c})$ method we use gene expression data for which an external evaluation criterion or class labels are available for the genes or samples or both.

**The Yeast cell data:** The Yeast cell data (Cho et al., 1988) show the fluctuation of the expression levels of approximately 6000 genes over two cycles (17 time points in total). Yeung and Raftery et al., (2001) used a 237 gene subset corresponding to four groups to illustrate clustering their method. This data subset is available at http://faculty.washington.edu/kayee/cluster/.

**The colon cancer data set:** The original data which contains over 6500 human genes, was first published by Alon et al., (1999). Following Alon et al., (1999), Getz et al. (2000) used a subset of these data containing the expression profile of 2000 genes in 22 normal tissues and 40 colon tissues. They normalized and organized these data as a $2000 \times 62$ expression level matrix. These data are available at http://microarray.princeton.edu/oncology/affydata/index.html

No external criteria for the classification of the genes is available for these data. A detailed analysis of these data is provided by Getz et al. (2000). Their analysis does not, however, give an estimate of the number of clusters present in the 2000 genes.

**The leukemia data set:** The data consist of 7,129 genes in 47 acute lymphoblastic (ALL) and 25 acute myeloid leukemia (AML) samples. These data are available at
6.2.2 Data Analysis

The main focus of these analyses is to discover the clusters present in the genes, or samples, or both, of microarray data described above via our method $MJD(\hat{c})$.

Using our method $MJD(\hat{c})$, we successfully estimated the “true” number of groups or clusters in the 237 gene subset of the yeast cycle data used by Yeung and Fraley et al. (2001). This estimate (peak of the curve) is depicted in Figure 6.1(a). Note that in the above estimation, the clustered objects (i.e the 237 genes) have dimension $p = n_s = 17$. By taking the 17 samples as $n = 17$ objects to be clustered, with the 237 genes as $p = 237$, we were able to correctly estimate the number of groups in the samples as two. See Figure 6.1(b).

The model-based approach of Fraley and Raftery (1998), however, failed to detect the correct number of groups in the genes. This is clearly depicted in Figure 6.1(c). The model-based approach could not cluster samples because this instance falls under the “large $p$, small $n$” paradigm. It is well known that model-based approaches have computational problems in such situations.

Our proposed method $MJD(\hat{c})$ also correctly estimates the correct cluster number (of the samples) in the colon cancer data. The cluster number estimates of both the samples and the genes of the colon cancer data (as indicated by the peaks of the curves) are respectively depicted in Figure 6.2(a) and Figure 6.2(b). $MJD(\hat{c})$
estimates the cluster number of the genes as eight. Although, not shown here, \( FR \) estimates the cluster number of the genes as nine. No class labels for gene are available for these data.

Using \( MJD(\hat{c}) \), we estimate the number of clusters in samples and the genes of the leukemia data, respectively as two and three. See Figure 6.3. The samples of these data are known to contain two natural groups: acute lymphoblastic (ALL) and acute myeloid leukemia (AML). No external criterion or labeling is available for the classification of the genes in these data.
Figure 6.1: Clustering the yeast cell data set (Yeung et al., 2001): (a) Clustering genes via $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (b) Clustering samples via $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (c) Clustering the genes via $FR$, a plot of BIC versus number of clusters $g$. 
Figure 6.2: Clustering the colon data set via $MJD(\hat{c})$: (a) Clustering samples, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$; (b) Clustering genes, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$. 

105
Figure 6.3: Clustering the leukemia data set via $MJD(\hat{c})$: (a) Clustering samples, plot of $W(g, \hat{c})$ versus number of clusters $g$; (b) Clustering genes, plot of $\hat{W}(g, \hat{c})$ versus number of clusters $g$. 

106
Chapter 7

Applications

In this Section we apply all three of our proposed methods in clustering two familiar real data sets.

7.1 Iris Dataset

This data set consists of four measurements of 50 plants from each of 3 species of Iris: Iris setosa, Iris versicolor, and Iris virginica. The measurements include sepal length, sepal width, petal length and petal width. The well known iris data first appeared in Anderson (1935) and were subsequently published by Fisher (1936) in his paper introducing the linear discriminant function. Following Fisher’s 1936 publication, the iris data became a standard benchmark set for the illustration of new multivariate techniques. As shown by Figure 7.1, the setosa plants (square points) are well separated from the versicolor (triangular points) and virginica plants (circular points). However, there is a clear overlap between the virginica and versicolor plants. The non-spherical
nature of these clusters coupled with the overlap between the virginica and versicolor plants makes the retrieval of the “natural” groups or clusters in these data a very difficult one for many clustering methods. The pairs plot of all the measurements is shown in Figure 7.2. We apply all three of our proposed methods to these data in order to illustrate their performance. The results are shown in Figure 7.3. Note that all three methods estimated the number of clusters in the Iris data as two. Many authors includ-
Figure 7.2: Pairs plot of the Iris data
ing Sugar and James (2003), suggest that even though the Iris data is known to contain three classes, from a clustering rather a classification perspective it may be sensible to consider the data as having two clusters consisting of one small cluster and a large one.

Figure 7.4 represents the model-based method of Fraley and Raftery (1998). This method chose 2 as the optimal number of clusters in the Iris data as indicated by highest peak of all six BIC curves in Figure 7.4. This method also chose Model six of Table 2.2 as best covariance structure of the Iris data. Model six of Table 2.2 assumes that the clusters in the Iris data are ellipsoidal, with variable volume and orientation but with the same shape.

Interesting results have been reached by the many authors who have worked on the Iris data. For example, Kryzanowski and Lai (1988) predict the optimal number of clusters embedded in the iris data as four. Their analysis also revealed two as a suboptimal cluster number. Sugar and James (2003) in their analysis, argue that the effective dimension of the iris data as obtained via principal component analysis is two. Proceeding on this assumption, they estimated the optimal cluster number as three.

Many authors including Fraley and Dean (2004) argue that the underlying structure or natural groupings of a multivariate data set may be contained in only a subset of the available variables. As such, the inclusion of unnecessary variables may often degrade the cluster number solutions of clustering methods. For instance, Raftery and Dean (2004) identified sepal length as an unnecessary variable (via their model-based variable selection procedure) in predicting the class structure of the Iris data. We shall refer to the Iris data without the sepal length feature as Iris3. Applying our
Figure 7.3: Cluster number estimates of the Iris data via all three of our proposed methods: (a) $MJ(\hat{c})$, a plot of $\hat{H}(g, \hat{c})$ versus $g$; (b) $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus $g$; (c) $MJD(0)$, a plot of $\hat{U}(g)$ versus $g$. 
Figure 7.4: BIC versus cluster number
Figure 7.5: Estimate of the cluster number of the iris data without sepal length
proposed method \( MJD(\hat{c}) \) to the Iris3 data we estimated the optimal cluster number as three. This is depicted in Figure 7.5. However methods \( MJ(\hat{c}) \) and \( MJD(0) \) still chose two as the optimal cluster number in the Iris3 data.

Next, we investigate how well the partitions induced by our methods agree with the natural known classes of the Iris data, via the “measures of agreement” discussed in Chapter 4. As shown in Table 7.1, by dropping the sepal length feature from the Iris data, our proposed method \( MJD(\hat{c}) \) is able to predict the cluster membership of the Iris data with greater accuracy. Note however that the “agreement” of the partitions induced by methods \( MJ(\hat{c}) \) and \( MJD(0) \) to the known class structure of the Iris data remain unchanged, even upon dropping the sepal length. Note that the Jaccard index estimates tend to be overly conservative, while those of the Rand index tend to be liberal.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Cluster no. est.(Method)</th>
<th>“Agreement Measure”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>( g = 2 ) (All) ( g = 3 ) (None)</td>
<td>0.76 ( \hat{g} = 2 ) (All) 0.86 ( \hat{g} = 3 ) (None)</td>
</tr>
<tr>
<td></td>
<td>Rand</td>
<td>0.74 AdRand 0.79 Jaccard 0.65 FM 0.75</td>
</tr>
<tr>
<td>Iris3</td>
<td>( g = 2 ) (( MJ(\hat{c}) ), ( MJD(0) )) ( g = 3 ) (( MJD(\hat{c}) ))</td>
<td>0.77 Rand 0.74 AdRand 0.59 Jaccard 0.76</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.94 0.91 0.84 0.91</td>
</tr>
</tbody>
</table>
7.2 Ruspini Data

The Ruspini data were first used by Ruspini (1970) to illustrate fuzzy clustering techniques. The data are two dimensional and consist of 75 points containing four clusters. These data are displayed in Figure 7.6(a). We use these data to illustrate the robustness of our proposed clustering methods to the presence of noise features/dimensions.

The cluster number estimates (as indicated by the peaks of the curves) obtained via all three of our proposed methods are depicted in Figures 7.6(b), (c) and (d). Not only did all three of our proposed methods, $MJ(\hat{c})$, $MJD(\hat{c})$ and $MJD(0)$, correctly predict the “true” cluster number ($g_0 = 4$) in these data, they also predicted cluster membership without error.

To demonstrate the robustness of our proposed methods to the presence of noise features we add $p_n$ extra noise features to the Ruspini data. We will refer to the resulting data as the modified Ruspini data. These noise features were independently generated from the standard normal distribution. We organize the modified Ruspini data as a $75 \times (2 + p_n)$ matrix.

Using $MJ(\hat{c})$ we were able to retrieve the right cluster number in the modified Ruspini data with $p_n = 1, 5$ and 10; see Figures 7.9(a), (b) and (c). We note that the addition of $p_n$ extra noise features to the Ruspini data did not affect the shape of the estimated maximal jump curve $\hat{H}(g, \hat{c})$ of original Ruspini data depicted in Figure 7.6(b). Figure 7.10 and Figure 7.11 respectively depict the estimates of the cluster number of
Table 7.2: Estimated Number of Clusters for Ruspini Data Sets

<table>
<thead>
<tr>
<th>No. of Added Noise Dimensions</th>
<th>Method</th>
<th>SJ</th>
<th>FR</th>
<th>MJ((\hat{c}))</th>
<th>MJD((\hat{c}))</th>
<th>MJD(0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

the modified Ruspini data via methods $MJD(\hat{c})$ and $MJD(0)$. Like $MJ(\hat{c})$, methods $MJD(\hat{c})$ and $MJD(0)$ tend to be robust to the presence of noise features. It is worth mentioning that, unlike our proposed methods, both $SJ$ and $FR$ methods can be greatly influenced by the presence of noise features. See Figures 7.7 and 7.8. In Figure 7.7(a), we note that even though $\hat{g}_{SJ} = 9$ for the Ruspini data with $p_n = 1$, the peak at $g = 4$ the “true” cluster number is also large. However as depicted by Figures 7.7(b) and (c), as $p_n$ increases, $SJ$ tend to estimate the cluster number as nine. The cluster number estimates of all three of our proposed methods together with those of $SJ$ and $FR$ are for the modified Ruspini data appear in Table 7.2.
Figure 7.6: Ruspini data:  
(a) Plot of the Ruspini data set;  
(b) Cluster number estimate via $MJ(\hat{c})$, a plot of $\hat{H}(g, \hat{c})$ versus cluster number $g$;  
(c) Cluster number estimate via $MJD(\hat{c})$, a plot of $\hat{W}(g, \hat{c})$ versus cluster number $g$;  
(d) Cluster number estimate via $MJD(0)$, a plot of $\hat{U}(g)$ versus cluster number $g$.  

117
Figure 7.7: Cluster number estimate of the modified Ruspini data with $p_n$ extra “noisy” features via the $SJ$ method: Plots of estimated jump $\hat{J}_g(p/2)$ versus cluster number $g$, with $p = 2 + p_n$ where, (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 
Figure 7.8: (a) Cluster number estimate of the modified Ruspini data with \( p_n \) extra “noisy” features via the FR method: Plots of BIC versus cluster number \( g \) (a) \( p_n = 1 \); (b) \( p_n = 5 \); (c) \( p_n = 10 \).
Figure 7.9: Cluster number estimate of the modified Ruspini data with $p_n$ extra “noisy” features via our proposed $MJ(\hat{c})$ method: Plots of $\hat{H}(g, \hat{c})$ versus cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 
Figure 7.10: Cluster number estimate of the modified Ruspini data with $p_n$ extra “noisy” features via our proposed MJD($\hat{c}$) method: Plots of $\hat{W}(g, \hat{c})$ versus cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 

121
Figure 7.11: Cluster number estimate of the modified Ruspini data with $p_n$ extra “noisy” features via our proposed $MJD(0)$ method: Plots of $\hat{U}(g)$ versus cluster number $g$ (a) $p_n = 1$; (b) $p_n = 5$; (c) $p_n = 10$. 
Chapter 8

Conclusion and Discussions

In this dissertation we focus mainly on developing methods for estimating the number of “natural” groups or clusters present in high dimensional data sets. We now summarize our contributions.

In chapter 1, we provide a survey of commonly used methods in estimating the cluster number in a data set. This survey is not meant to be comprehensive, but rather aimed at grouping these methods along similar motivations. The first category of methods tend to formulate the cluster number estimation problem as choosing the number of components in a finite mixture model. The second is the hypothesis testing approach. Methods under this category attempts to solve this problem by deriving formal tests of the cluster number. The problem with this approach is that most valid tests tend to have intractable sampling distributions or involve hypotheses for which the rejection is uninformative. The third category of methods estimates the “optimal” number of clusters in a data set based on procedures that are largely heuristic but
intuitively reasonable. Most commercial software for clustering are of this last kind.

Our major contribution is in chapter 2. First, the ineffectiveness of the estimated distortion curve (a plot of \( \hat{d}_g \) versus \( g \)) for assessing the number of clusters in a data set is demonstrated. In its place, we proposed a new quantity, namely, the maximal jump function, \( H(g, c) \). The maximal jump function measures the excess transformed distortion attainable by fitting an extra cluster to a data set. Large values of the estimated maximal jump function \( \hat{H}(g, \hat{c}) \) tend to suggest optimality at \( g \). Some theoretical properties of the maximal jump function \( H(g, c) \) have also been discussed.

Based on the maximal jump function \( H(g, c) \), three stopping rules were proposed for determining the number of clusters in data: \( MJ(\hat{c}) \), \( MJD(\hat{c}) \) and \( MJD(0) \). We note that \( MJ(\hat{c}) \) and \( MJD(\hat{c}) \) require the pre-specification of the scalar \( \hat{c} \). Preliminary simulations have shown that the choice of \( \hat{c} \) can “significantly” affect the results of \( MJ(\hat{c}) \). However, \( MJD(\hat{c}) \), which is the best method among the three stopping rules, tends to be more robust to the choice of \( \hat{c} \), provided \( \hat{c} \) is “sufficiently” less than \( \hat{d}_{g_{\max}} \).

The choice \( \hat{c} = \hat{d}_{2g_{\max}} \), which was used for all our simulations, tends to provide good results. We tested the above mentioned stopping rules via Monte Carlo simulations in chapter 3. Also, to show the effectiveness of these stopping rules, comparisons were made to the well known model-based clustering approach of Fraley and Raftery (1998) and to the information theory approach of Sugar and James (2003).

Factors such as the dimension \( p \) of data, the presence of noise dimensions in the data, the total number of points \( n \) in the data, cluster orientation, the underlying distribution of data, etc., tend to affect the overall performance of clustering methods.
The importance of these factors may vary from method to method. In Chapter 4, we employed an analysis of variance type analysis to identify important factors. The most important factors are dimension $p$ of a data set, the number of “important” dimensions $p_I$ in the data set, the covariance structure of real data, and how well separated the cluster centers are from each other as measured by the cluster separation index $SEPINDEX$.

Clustering data sets that tend to possess highly correlated features/dimensions can be difficult for many methods. In Chapter 5, we proposed a clustering procedure for data sets possessing highly correlated features/dimensions with roughly the same covariance structure within each group. This procedure tends to be a great improvement over $MJD(\hat{c})$ for the above kind of data sets. In Chapter 6, we tested $MJD(\hat{c})$ on simulated data sets that fall under the “large $p$, small $n$” paradigm as well as DNA microarray data sets, with encouraging results. Here, $MJD(\hat{c})$ is also compared to the prediction strength method of Tibshirani and Walther(2005). Finally, we applied all our proposed methods in clustering the well known Iris data and the Ruspini data in Chapter 7.

A major advantage of our proposed methods is that, unlike many methods, they tend to be robust to the presence of noise features/dimensions. They have also been applied successfully in clustering data sets in which the dimension $p$ far exceeds the the sample size $n$. Our proposed method $MJD(\hat{c})$, in particular, has shown great promise in clustering DNA microarray data.

Our proposed methods tend to perform better when the number of informative
dimensions is large. This property is consistent with the description of Krzanowski and Lai (1988) of a good clustering method. Krzanowski and Lai (1988) assert that a good clustering criterion is one that capitalizes on extra information and exhibits greater precision with increased dimensionality, while a poor method tends to remain fairly constant irrespective of dimension. Also, our proposed methods perform reasonably well for small $p$ if the clusters are moderately spaced. For high-dimensional data, our proposed methods tend to outperform the model-based method of Fraley and Raftery (1998).

We conclude by noting that the estimation of the number of “natural” groups in a data set is crucial and indispensable in many clustering applications. However, up to now, no solution that allows general applicability to real life problems is available to what has come to be known as the cluster number problem. Noise features/dimensions are commonplace in many applications. The real structure of data is often contained only in a small subset of its features/dimensions (Raftery and Dean, 2005). The problem of determining which subset is informative is, however, not an easy task with high-dimensional data. It is worth noting that in an attempt to find the desired subset, the real structure or the data-generating process of the data could be destroyed. In this regard, clustering methods that can unravel the real structure of data embedded in noisy features/dimensions without altering the number of features will be desirable. Most clustering methods tend not to be robust to the presence of noise features. Our proposed methods exhibit robustness in clustering data with large number of noise features/dimensions.


129


Appendix
Appendix A

Proof of Proposition 2.4.1

Proposition 2.4.1

Define $d_0^{-t} \equiv 0$. Let $c > 0$ and consider the jump function

$$J(g, t, c) = \left( \frac{d_g}{c} \right)^{-t} - \left( \frac{d_{g-1}}{c} \right)^{-t}, \quad t \in (0, \infty), \quad g = 1, \ldots, g_{\text{max}}, \quad (A.1)$$

where $d_g$ is the distortion for $g$ clusters. If $d_g > c$ for all $g$, then,

$$\tau_g = \frac{\ln \left( \frac{\ln d_g - \ln c}{\ln d_{g-1} - \ln c} \right)}{\ln \left( \frac{d_g}{d_{g-1}} \right)},$$

satisfies

$$J(g, t, c) \leq J(g, \tau_g, c), \quad \text{for all } t \in (0, \infty).$$

Furthermore, if in addition, $d_1 > d_2 > \cdots > d_{g_{\text{max}}}$, then

(a) $\lim_{d_g \to c} \tau_g = \infty$,

(b) $\tau_2 < \cdots < \tau_{g_{\text{max}}}$.
(c) \(0 < \tau_g < 1\) for all \(g\) provided \(c^{-1}d_g > \exp(1)\) for all \(g\).

**Proof:**

Let \(c > 0\) and consider the jump function

\[
J(g, t, c) = \left(\frac{d_g}{c}\right)^{-t} - \left(\frac{d_{g-1}}{c}\right)^{-t}, \quad t \in (0, \infty), \; g = 1, \ldots, g_{\text{max}}. \tag{A.2}
\]

For a fixed \(g\), let \(k_1 = \ln \left(\frac{d_g}{c}\right)\), and \(k_2 = \ln \left(\frac{d_{g-1}}{c}\right)\).

Now define \(A(t) = J(g, t, c)\), so that (A.2) becomes

\[
A(t) = \exp(-k_1 t) - \exp(-k_2 t), \quad t > 0, \; 0 < k_1 < k_2. \tag{A.3}
\]

We wish to show that there exists \(\tau_g\) such that

\[
A(t) \leq A(\tau_g), \quad \text{for all } t > 0. \tag{A.4}
\]

That is, \(\tau_g\) is a global maximum of \(A(t)\). To prove that \(\tau_g\) is a global maximum of \(A(t)\), it is sufficient to establish the following:

1. \(\tau_g\) is the only critical point of \(A(t)\);
2. Show that \(\tau_g\) is a local maximum of \(A(t)\);
3. \(\lim_{t \to 0} A(t) = 0\) and \(\lim_{t \to \infty} A(t) = 0\).

From (A.3), we note the following:

\[
A'(t) = -k_1 \exp(-k_1 t) + k_2 \exp(-k_2 t). \tag{A.5}
\]
\[ A''(t) = k_1^2 \exp(-k_1 t) - k_2^2 \exp(-k_2 t). \]  
(A.6)

Now note that if \( \tau_g \) exists, it must be a critical point of \( A(t) \). That is

\[ A' (\tau_g) = 0. \]  
(A.7)

On solving (A.7), we obtain

\[ \tau_g = \frac{\ln \left( \frac{k_1}{k_2} \right)}{k_1 - k_2}, \]  
(A.8)

as the only critical point.

Next, we show that \( \tau_g \) is indeed a local maximum of \( A(t) \). To do this, it is sufficient to show \( A''(\tau_g) < 0 \). By putting (A.8) into (A.6), we obtain

\[ A''(\tau_g) = k_1^2 \left( \frac{k_1}{k_2} \right)^{k_1 - k_2} - k_2^2 \left( \frac{k_1}{k_2} \right)^{k_1 - k_2}. \]  
(A.9)

Now consider

\[ A''(\tau_g) = k_1^2 \left( \frac{k_1}{k_2} \right)^{\frac{k_1}{k_1 - k_2}} - k_2^2 \left( \frac{k_1}{k_2} \right)^{\frac{k_2}{k_1 - k_2}} \]  
(A.10)

\[ = \left( \frac{k_1}{k_2} \right)^{\frac{k_1}{k_1 - k_2}} \left[ k_1^2 - k_2^2 \left( \frac{k_1}{k_2} \right)^{\frac{k_2}{k_1 - k_2}} \left( \frac{k_1}{k_2} \right)^{\frac{k_1}{k_1 - k_2}} \right] \]  
(A.11)

\[ = \left( \frac{k_1}{k_2} \right)^{\frac{k_1}{k_1 - k_2}} [k_1^2 - k_2 k_1] \]  
(A.12)

\[ = \left( \frac{k_1}{k_2} \right)^{\frac{k_1}{k_1 - k_2}} k_1 |k_1 - k_2|. \]  
(A.13)

Clearly, from (A.13), \( A''(\tau_g) < 0 \), since \( 0 < k_1 < k_2 \), thus establishing \( \tau_g \) as a local maximum of \( A(t) \). By substituting \( k_1 = \ln (d_g/c) \) and \( k_2 = \ln (d_{g-1}/c) \) into (A.8) we
We note from (A.14), \( \tau_g \) exists for all \( g \in \{2, \ldots, g_{\text{max}}\} \) provided \( d_g > c \) for all \( g \).

Also from (A.3), we note that \( \lim_{t \to 0} A(t) = 0 \) and \( \lim_{t \to \infty} A(t) = 0 \).

Hence, \( \tau_g \) is a global maximum of \( A(t) \) and exists provided \( d_g > c \) for all \( g \).

(a) Since \( d_{g-1} > d_g \) for all \( g \), for a fixed \( g \), we may write

\[
\ln d_{g-1} = \ln d_g + \epsilon, \tag{A.15}
\]

for some \( \epsilon > 0 \) where \( \epsilon \) depends on \( g \).

By substituting (A.15) into (A.14), we obtain

\[
\tau_g = \frac{\ln \left[ \frac{\ln d_g - \ln c}{\ln d_{g-1} - \ln c} \right]}{\ln \left( \frac{d_g}{d_{g-1}} \right)}. \tag{A.16}
\]

\[
= \frac{\ln \{1 + \epsilon/\ln (d_g/c)\}}{\epsilon} \tag{A.17}
\]

\[
\to \infty, \tag{A.18}
\]

as \( d_g \to c \).
(b) Let $\theta_g = \ln \left( \frac{d_g}{c} \right)$, $\theta_{g-1} = \ln \left( \frac{d_{g-1}}{c} \right)$ and $\theta_{g-2} = \ln \left( \frac{d_{g-2}}{c} \right)$. Then

$$
\Delta_g = \tau_g - \tau_{g-1} = \ln \left( \frac{\theta_g}{\theta_{g-1}} \right) - \ln \left( \frac{\theta_{g-1}}{\theta_{g-2}} \right).
$$

Note that $\theta_g < \theta_{g-1} < \theta_{g-2}$, $g \geq 3$. For some $\delta, \varepsilon > 0$, let $\theta_{g-1} = \theta, \theta_g = \theta - \delta$, $\theta_{g-2} = \theta + \varepsilon$. Thus,

$$
\Delta_g = \frac{\ln \left( \frac{\theta - \delta}{\theta} \right) - \ln \left( \frac{\theta}{\theta + \varepsilon} \right)}{(\theta - \delta) - \theta} - \frac{\ln \left( \frac{\theta}{\theta + \varepsilon} \right)}{\theta - (\theta + \varepsilon)}.
$$

(A.19)  

By the mean value theorem, we may write (A.20) as

$$
\Delta_g = \frac{1}{\theta - \delta^*} - \frac{1}{\theta + \varepsilon^*},
$$

(A.21)

where $0 < \delta^* < \delta$ and $0 < \varepsilon^* < \varepsilon$.

Note from (A.21) that $\Delta_g > 0$, since $\theta > 0$ and $\theta - \delta^* > 0$. Therefore, $\tau_{g-1} < \tau_g$ for all $g \geq 3$.

(c) To prove $0 < \tau_g < 1$ for all $g$ provided $c^{-1}d_g > \exp(1)$ for all $g$, it suffices to show that

$$
0 < \frac{\ln (\Delta + \theta_g) - \ln (\theta_g)}{\Delta} < 1,
$$

where $\theta_g = \ln \left( \frac{d_g}{c} \right)$, $\theta_g > 1$ and $\Delta = \ln \left( \frac{d_{g-1}}{c} \right) - \ln \left( \frac{d_g}{c} \right)$.

Clearly, $\frac{\ln (\Delta + \theta_g) - \ln (\theta_g)}{\Delta} > 0$ since $\theta_g > 1$. By the mean value theorem, there
exists $\Delta^*$, $0 < \Delta^* < 1$ such that

$$\frac{\ln (\Delta + \theta_g) - \ln (\theta_g)}{\Delta} = \frac{1}{\Delta^* + \theta_g} < 1.$$
Appendix B

Derivation of Equation (2.17)

Let \( c > 0 \) and define

\[
U(g) = \lim_{c \to 0} (-\ln c) [W(g,c)],
\]

where

\[
W(g,c) = H(g,c) - H(g+1,c)
\]

and \( H(g,c) \) is the maximal jump function associated with \( g \) clusters defined by

\[
H(g,c) = J(g,\tau_g,c),
\]

where \( J(g,t,c) \leq J(g,\tau_g,c) \), for all \( t > 0 \) and \( J(g,t,c) \) is defined by (A.2).

Here show that

\[
U(g) = \exp(-1) \left[ \ln \left( \frac{d_{g-1}}{d_g} \right) - \ln \left( \frac{d_g}{d_{g+1}} \right) \right].
\]  \hspace{1cm} (B.1)
Derivation of Equation (2.17):

Consider the maximal jump function

\[ H(g, c) = J(g, \tau_g, c) \]

\[ = c^\tau_g \left[ \exp (-a\tau_g) - \exp (-b\tau_g) \right], \]

(B.3)

where \( a = \ln d_g, b = \ln d_{g-1} \).

Note that \( \tau_g \) is given by

\[ \tau_g = \arg \max_{t>0} J(g, t, c) \]

\[ = \frac{\ln \left( \frac{d_g - \ln c}{\ln d_{g-1} - \ln c} \right)}{\ln \left( \frac{d_g}{d_{g-1}} \right)}, \text{ for } g = 2, 3, \ldots \]

Further note that we may rewrite \( \tau_g \) as

\[ \tau_g = \frac{\ln \left( \frac{a - \ln c}{b - \ln c} \right)}{a - b}, \]

(B.4)

\[ = \frac{\ln \left( 1 + \frac{a-b}{b - \ln c} \right)}{a - b}. \]

(B.5)

Thus
\[
\lim_{c \to 0} \tau_g = \lim_{c \to 0} \left\{ \frac{\ln \left( 1 + \frac{a - b}{b - \ln c} \right)}{a - b} \right\} \\
= \ln \left( 1 + \lim_{c \to 0} \left\{ \frac{a - b}{b - \ln c} \right\} \right) \\
= \frac{\ln 1}{a - b} \\
= 0.
\]

By Young’s form of Taylor’s expansion

\[
\frac{\exp(-x) - (1 - x)}{x} \to 0 \text{ as } x \to 0.
\]

Thus, \( \exp (-a \tau_g) - \exp (-b \tau_g) = (b - a) \tau_g + R \tau_g \), where \( R \tau_g / \tau_g \to 0 \) as \( \tau_g \to 0 \).

Then

\[
(- \ln c) H(g, c) = (- \ln c) c^{\tau_g} \left[ \exp (-a \tau_g) - \exp (-b \tau_g) \right] \quad \text{(B.6)}
\]

\[
= (- \ln c) c^{\tau_g} \left[ (b - a) \tau_g + R \tau_g \right]. \quad \text{(B.7)}
\]

If \( (\ln c) \tau_g \to -1 \) as \( c \to 0 \), then \( c^{\tau_g} \to \exp(-1) \) as \( c \to 0 \) and

\[
\lim_{c \to 0} (- \ln c) H(g, c) = (b - a) \exp(-1) \\
= \exp(-1) \ln \left( \frac{d_{g-1}}{d_g} \right).
\]
Similarly, \( \lim_{c \to 0} (-c \ln c)H(g + 1, c) = \exp(-1) \ln \left( \frac{d_g}{d_{g+1}} \right) \), leading to the result

\[
U(g) = \lim_{c \to 0} \{- \ln c [H(g, c) - H(g + 1, c)]\}
\]

\[
= \exp(-1) \left[ \ln \left( \frac{d_{g-1}}{d_g} \right) - \ln \left( \frac{d_g}{d_{g+1}} \right) \right].
\]

To show \( (\ln c)\tau_g \to -1 \) as \( c \to 0 \), note that by Young’s form of Taylor’s expansion

\[
(a - b)\tau_g = \ln \left( 1 + \frac{a - b}{b - \ln c} \right) = \left( \frac{a - b}{b - \ln c} \right) - \frac{1}{2} \left( \frac{a - b}{b - \ln c} \right)^2 + R_c,
\]

where \( R_c/ [(a - b)/(b - \ln c)] \to 0 \) as \( c \to 0 \). Then \( (\ln c)\tau_g \to -1 \) as \( c \to 0 \) follows easily since \( (a - b)/(b - \ln c) \to 0 \) as \( c \to 0 \).