Variable Selection and Other Extensions of the Mixture Model Clustering Framework

Nema Dean

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2006

Program Authorized to Offer Degree: Statistics

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Nema Dean

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

_____

Adrian Raftery

Reading Committee:

_____

Adrian Raftery

_____

Matthew Stephens

_____

Werner Stuetzle

Date: _____

Signature_____

Date_____

University of Washington

Abstract

Variable Selection and Other Extensions
of the
Mixture Model Clustering Framework

Nema Dean

Chair of the Supervisory Committee:
Professor Adrian Raftery
Department of Statistics

Mixture model clustering has recently come to the forefront of techniques used in unsupervised learning. Its advantages include an intuitive model set-up (one cluster = one density), a statistical framework allowing model selection techniques to be utilized which can answer questions such as choosing the number of clusters believed to be present in the data and identifying (and modeling) outliers, and efficient algorithms for fitting models to data. In continuous data cases usually the normal distribution is assumed for the clusters, which has come to be called "model-based clustering". In discrete data cases, the multinomial or Bernoulli distributions are assumed for individual variables in clusters, with the additional assumption of conditional independence of variables given cluster membership giving a multivariate model which remains reasonably parsimonious, known as "latent class analysis".

Although the mixture model clustering framework is applicable in a wide range of situations, new demands due to the increasing complexity of data available require extensions of the framework in order for it to remain useful. In the past, datasets usually were expensive to assemble and so the variables recorded were few and carefully chosen. Now datasets can often have as many variables as observations, if not more,

not all of which are relevant. A principled method of selecting variables important to clustering is necessary since having more variables generally restricts the range of models possible to fit to the data and inclusion of noise variables can degrade the quality of clustering found and lead to the incorrect number of clusters being selected. In this thesis we present a stepwise model-based approach to variable selection particularly tailored to mixture model clustering in the context of model-based clustering and latent class analysis. We also look at somewhat the reverse problem of identifying differentially expressed genes (observations instead of variables) in cDNA microarray datasets using a mixture model approach with appropriate normalizations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank Professor Adrian Raftery for helping to guide and teach me so much. I cannot imagine having completed graduate school without his help. I would also like to gratefully acknowledge the financial support of grant R01 EB002137-02. The members of my committee, Werner Stuetzle and Matthew Stephens, have been very supportive and generous with their time and advice for which I am extremely grateful. The Model-Based Clustering Working Group has also been of great assistance, particularly Chris Fraley. My thanks (I think) also go to Dr. Brendan Murphy for putting me on the path of Statistics and towards the UW. Finally, an acknowledgment must be made of the tremendous debt I owe to: my parents John and Noëleen Dean, my sister Tanya Dean, my grandparents Catherine Hamill, John and Mary Dean and all my friends both Irish, American and Canadian, who have carried me to this point. Thank you.

Chapter 1

GENERAL OVERVIEW

1.1 Mixture Models and Clustering

Mixture models are a natural idea for extending single densities to a more complex and flexible form of modeling of data, the idea of which has been around for over 100 years ([60]).The basic concept is treating a population as being made up of several sub-populations and modeling each sub-population with its own density and the overall population as a weighted sum of these densities.

However this methodology was not truly feasible in practical terms until the advent of the EM algorithm, making finding maximum likelihood estimates viable ([20]). In the last 20 years mixture modelling has found an increasingly large number of applications. One of the most useful and interesting of which has been in clustering where the object of interest is the underlying (unknown) group structure.

Clustering, which is the discovery of unknown group structure in data, had been of interest prior to the introduction of the mixture model based approach and many algorithms and heuristic methodologies exist for this problem. However these approaches lack the statistical modeling framework which allows many important questions to be answered in a statistically principled way. Questions such as, the true number of mixture components needed, whether or not outliers are present in the data or not and what form the components should take, are examples of the questions answerable via the mixture model approach, part of the advantages inherent in this approach, not present in many of the previous heuristic approaches. These questions and their solutions are reviewed in the subsequent chapters. An excellent review of mixture

models in general is given by [53].

1.2 Variable Selection

Traditionally, datasets for clustering tended to be expensive to assemble with a limited number of observations and variables. So variables tended to be carefully chosen in advance to best display the heterogeneity in the data. Nowadays of course, the opposite is true, so much information can be and is collected that it is important to ensure that the information of interest in the relevant variables/features is not swamped beneath noise from the other variables.

Enough variables included with no information about the group structure can degrade the performance of most estimating techniques in supervised and unsupervised learning. While much work has been done to address the problem of variable/feature selection in supervised learning, less work has been done in the unsupervised learning context.

Although the degradation of estimated group structures found in the presence of too many noise variables is of course important, equally important is the re-occurring problem of choice of number of components, since this can also be affected by the inclusion of noise variables.

Another issue is the limiting effect of large numbers of variables on the range and type of clustering models available to fit to the data and to choose from. If we have more variables than observations, hardly any models with group structure can be fit to the data. Even if the number of observations is larger than variables, sparsity of the data can still mean only models with a small number of components are identifiable.

In chapter 2 of this thesis we introduce the idea of mixture modeling and its application to clustering in the continuous data setting, called model-based clustering and present two models for evaluating a variable's clustering contribution along with a greedy search method used to explore the model space. The models give two alternative ways of splitting the density of all variables (conditional on the clus-

ter membership variable) into conditional densities involving three sets of variables : those already selected for clustering, the variable under consideration and all other variables (a possibly high-dimensional set). The model assuming the variable under consideration is not useful for clustering, given the clustering variables already selected, allows this variable to be conditionally independent of the cluster membership variable given the clustering variables. The clustering variables are always dependent on the cluster membership variable. In the other model this conditional independence is not allowed and both clustering variables and the variable under consideration are dependent on the cluster membership variable. Comparing the fit of these two models to the data allows us to make a decision of whether to include the variable under consideration in the set of clustering variables or not.

In chapter 3 we introduce another specific form of mixture model clustering, this time for discrete data, called latent class analysis. In this form of mixture model clustering, variables are assumed to be conditionally independent given the cluster membership variable and individual variables are modeled with multinomial or Bernoulli densities. Again we present two alternative models for checking a variable's clustering contribution along with a different search method for exploring the model space (based on the headlong search [4]). This time the model assuming the variable under consideration is not useful for clustering, given the clustering variables already selected, allows this variable to be fully independent of the clustering variables as well as of the cluster membership variable. The clustering variables are always dependent on the cluster membership variable. In the other model this independence is not allowed and both clustering variables and the variable under consideration are dependent on the cluster membership variable (but are still conditionally independent of one another as in the model assumption for latent class models). Comparing the fit of these two models to the data allows us to make a decision of whether to include the variable under consideration in the set of clustering variables or not.

1.3 cDNA Microarray Gene Expression Data

One of the prime examples of data that requires methodology beyond that currently implemented in traditional supervised and unsupervised learning techniques is gene expression data. This microarray technology allows for the simultaneous recording of expression levels of thousands of genes/gene segments under numerous different conditions ([63]. One glitch in the analysis of data resulting from this technology is the lamentable lack of replication in most datasets and the large quantity of noise that is therefore difficult to filter out (often making strong distributional assumptions necessary). Another more fundamental issue is the fact that n, the number of observations/experiments/conditions is typically much, much smaller than p, the number of variables/genes. This is the *exact reverse* of the traditional data structure where $n >> p$.

In the typical technical replication type of experiment, cDNA from each condition is labeled with either a red or green dye and both are hybridized to a slide with cDNA of genes or gene sections of interest on it and the expression levels for the genes in each condition are extracted from the slide. Although the labeling scheme may sometimes change from slide to slide or experiment set-up to experiment set-up, e.g. in one set-up, control condition labeled red and treatment labeled green, and in another, control condition labeled green and treatment labeled red, each condition has its own dye for each slide. In the set-up for biological replication types of experiment, cDNA from each condition is labeled with the same dye and compared to a reference sample labeled with the other dye. For example, control and treatment cDNA may both be labeled with green dye while the reference sample is labeled with red dye and each experiment consists of a hybridization of the reference sample and one of either the control or treatment samples to a slide.

1.4 Differential Expression Detection

Another way in which mixture model methodology can be useful for answering scientific questions is looking at the issue of differential expression detection in microarrays. One of the more interesting biological questions for gene expression data of either type of experimental set-up is are the expression levels for a particular gene (or set of genes) different under different conditions. For example it may be of interest to know which set of genes have higher levels of expression in cancer tissues samples, say than in healthy tissue samples, as this information may be useful for diagnosis, prognosis or treatment of patients in the future. Genes that have higher levels of expression in treatment samples than in control samples are often called over-expressed and genes that have lower levels of expression in treatment samples than in control samples are often called under-expressed. Under- and over-expressed genes are the two types of *differentially expressed* genes.

An obvious test for each gene individually (if we have replicates/repeated measurements of expression levels) is a t-test of the (average of the) logged control versus treatment expression ratios for the null hypothesis of true mean of zero versus non-zero. However, since there are usually thousands of genes to be tested in this way, multiple testing issues come into play. Another issue, in addition to the problem of multiple testing, is the small amount of replicates (if any) generally available for estimating the standard deviation for each t-test. If we can estimate the standard deviation, the low number of replicates usually present means that estimate is extremely variable. While it may be possible to borrow strength from other genes in estimating the standard deviation it is clearly not correct to use both differentially and non-differentially expressed genes to estimate this.

In chapter 4 of this thesis we present a model for identifying differentially expressed genes based on the idea of modeling outliers in continuous data by a uniform distribution in a mixture model (see [5, 67]). We posit that the measurements for

differentially expressed genes represent outliers from the main distribution of non-differentially expressed genes' measurements. The (average) log ratios of expression levels of non-differentially expressed genes are modeled by a normal distribution while those of differentially expressed genes are modeled with a uniform distribution. Different normalizations prior to fitting the model are presented for different types of data: single and multiple replicate, and distinct comparison experiment set-up types.

1.5 Contributions

- Proposing a principled method of selecting variables important to clustering using two different models for the same set of variables, split into different conditional distributions involving mixtures

- Tailoring the models to mixture model clustering in the context of model-based clustering

- Tailoring the models to mixture model clustering in the context of latent class analysis

- Proposing and developing search algorithms for searching the space of models and variables

- Presenting a simple model of a mixture of uniform (for differentially expressed genes) and normal (for non-differentially expressed genes) distributions to identify differentially expressed genes via conditional probability of membership for each gene being in the uniform mixture model component

- Presenting extensions of the mean normalization in [26] for mean and variance normalization prior to modeling, which can also improve the performance of other methods of differential gene expression detection (e.g. the simple rule of two).

Chapter 2

VARIABLE SELECTION FOR MODEL-BASED CLUSTERING

2.1 Introduction

In classification, or supervised learning problems, the structure of interest may often be contained in only a subset of the available variables and inclusion of unnecessary variables in the learning procedure may degrade the results. In these cases some form of variable selection prior to, or incorporated into the fitting procedure may be advisable. Similarly, in clustering, or unsupervised learning problems, the structure of greatest interest to the investigator may be best represented using only a few of the feature variables. This may give the best clustering model to describe future data, or fewer variables may give a better partition of the data into clusters closer to the true underlying group structure. However, in clustering the classification is not observed, and there is usually little or no *a priori* knowledge of the structure being looked for in the analysis, so there is no simple pre-analysis screening technique available to use. In this case it makes sense to consider including the variable selection procedure as part of the clustering algorithm.

In this chapter, we introduce a method for variable or feature selection for model-based clustering. The basic idea is to recast the variable selection problem as one of comparing competing models for all the variables initially considered. Comparing two nested subsets is equivalent to comparing two models, in one of which all the variables in the bigger subset carry information about cluster membership, while in the other model the variables considered for exclusion are conditionally independent of cluster membership given the variables included in both models. This comparison is

made using approximate Bayes factors. This model comparison criterion is combined with a greedy search algorithm to give an overall method for variable selection. The resulting method selects the clustering variables, the number of clusters, and the clustering model simultaneously.

The variable selection procedure suggested in this chapter is tailored specifically for model-based clustering and, as such, incorporates the advantages of this paradigm relative to some of the more heuristic clustering algorithms. They include an automatic method for choosing the number of clusters, only one user-defined input necessary (the maximum number of clusters to be considered) that is easily interpretable, and a basis in statistical inference.

A brief review of mixture model clustering is given in Section 2.2.1. The statistical models behind the variable selection method are explained in Section 2.2.2 and the greedy search algorithm is introduced in Section 2.2.3. In Section 2.2.4, the specific example of clustering with Gaussian components (model-based clustering) and allowing different covariance formulations is discussed. Results comparing the performance of model-based clustering with and without variable selection are given in Section 2.3 for simulated data and in Section 2.4 for some real data examples. The advantages and limitations of the method are discussed in Section 2.5, which also mentions some other work on the problem. Finally, the greedy search clustering variable selection procedure steps are discussed in greater detail in the appendix.

2.2 Methodology

2.2.1 Mixture Model Clustering

Mixture model clustering is based on the idea that the observed data come from a population with several subpopulations. The general idea is to model each of the subpopulations separately and the overall population as a mixture of these subpopulations, using finite mixture models. Mixture model clustering goes back at least to

[71] and reviews of the area are given by [53] and [30].

The general form of a finite mixture model with G subpopulations or groups is

$$f(\mathbf{x}) = \sum_{g=1}^{G} \pi_g f_g(\mathbf{x}),$$

where π_g is the proportion of the population in the gth group, and $f_g(\cdot)$ is the probability density function for the gth group. The subpopulations are often modeled by members of the same parametric density family, in which case the finite mixture model can be written

$$f(\mathbf{x}) = \sum_{g=1}^{G} \pi_g f(\mathbf{x}|\phi_g),$$

where ϕ_g is the parameter vector for the gth group.

The mixture model can be used to partition the data into clusters using the optimal or Bayes rule for classification. This classifies observation $\mathbf{x}$ to cluster g if the posterior probability that it belongs to group g is greater than the posterior probabilities that it belongs to any other group, i.e. if $\tau_g(\mathbf{x}) > \tau_h(\mathbf{x})$, $h = 1, \ldots, G$, where $\tau_h(\mathbf{x}) = \pi_h f_h(\mathbf{x}) / \sum_{g=1}^{G} \pi_g f_g(\mathbf{x})$ is the posterior probability that it belongs to the hth group. Since the denominator is the same in all posterior probabilities, it is possible to simply define the Bayes rule as follows: classify $\mathbf{x}$ into cluster g if $g = \arg\max_h \pi_h f_h(\mathbf{x})$. We can approximate the Bayes rule by replacing the unknown parameters by their estimated values. This is called the plug-in rule. In our examples, we compare the partition given by the finite mixture model defined on the subset of selected variables to a known underlying classification, in order to assess how much improvement the variable selection procedure gives in clustering.

A difficulty of some of the more heuristic clustering algorithms is the lack of a statistically principled method for determining the number of clusters. Since it is an inferentially based procedure, mixture model clustering can use model selection methods to make this decision. Bayes factors [41] are used to compare the models. This permits comparison of the non-nested models that arise in this context.

The Bayes factor for a model M_1 against a competing model M_2 is equal to the posterior odds for M_1 against M_2 when their prior model probabilities are equal. It is computed as the ratio of the integrated likelihoods for the two models. This ratio can be hard to compute, and we use the easily calculated Bayesian information criterion (BIC) as the basis for an approximation. This is defined by

$$\begin{aligned} BIC &= 2 \times \log(\text{maximized likelihood}) \\ &- (\text{no. of parameters}) \times \log(n), \end{aligned} \tag{2.1}$$

where n is the number of observations. Twice the logarithm of the Bayes factor is approximately equal to the difference between BIC values for the two models being compared. We choose the number of groups and the parametric model by recognizing that each different combination of number of groups and parametric constraints defines a model, which can then be compared to others. [44] showed BIC to be consistent for the choice of the number of clusters in the case of Gaussian mixture with certain constraints. Differences of less than 2 between BIC values are typically viewed as barely worth mentioning, while differences greater than 10 are often regarded as constituting strong evidence [41].

2.2.2 Models for Variable Selection

To address the variable selection problem, we recast it as a model selection problem. We have a data set Y, and at any stage in our variable selection algorithm, it is partitioned into three sets of variables, $Y^{(clust)}$, $Y^{(?)}$ and $Y^{(other)}$, namely:

- $Y^{(clust)}$: the set of already selected clustering variables,
- $Y^{(?)}$: the variable(s) being considered for inclusion into or exclusion from the set of clustering variables, and
- $Y^{(other)}$: the remaining variables.

The decision for inclusion or exclusion of $Y^{(?)}$ from the set of clustering variables is then recast as one of comparing the following two models for the full data set:

$$\begin{aligned} M_1 : p(Y|\mathbf{z}) &= p(Y^{(clust)}, Y^{(?)}, Y^{(other)}|\mathbf{z}) \\ &= p(Y^{(other)}|Y^{(?)}, Y^{(clust)})p(Y^{(?)}|Y^{(clust)})p(Y^{(clust)}|\mathbf{z}) \qquad (2.2) \\ M_2 : p(Y|\mathbf{z}) &= p(Y^{(clust)}, Y^{(?)}, Y^{(other)}|\mathbf{z}) \\ &= p(Y^{(other)}|Y^{(?)}, Y^{(clust)})p(Y^{(?)}, Y^{(clust)}|\mathbf{z}), \end{aligned}$$

where $\mathbf{z}$ is the (unobserved) set of cluster memberships. Model M_1 specifies that, given $Y^{(clust)}$, $Y^{(?)}$ is conditionally independent of the cluster memberships (defined by the unobserved variables $\mathbf{z}$), that is, $Y^{(?)}$ gives no additional information about the clustering. Model M_2 implies that $Y^{(?)}$ does provide additional information about clustering membership, after $Y^{(clust)}$ has been observed.

An important aspect of the model formulation is that it does not require that irrelevant variables be independent of the clustering variables. If instead the independence assumption $p(Y^{(?)}|Y^{(clust)}) = p(Y^{(?)})$ were used in model M_1, we would be quite likely to include redundant variables that were related to the clustering variables but not to the clustering itself. We assume that the remaining variables $Y^{(other)}$ are conditionally independent of the clustering given $Y^{(clust)}$ and $Y^{(?)}$ and belong to the same parametric family in both models. The difference between the assumptions underlying the two models is illustrated in Figure 3.1, where arrows indicate dependency.

Models M_1 and M_2 are compared via an approximation to the Bayes factor which allows the high-dimensional $p(Y^{(other)}|Y^{(?)}, Y^{(clust)})$ to cancel from the ratio. The Bayes factor, B_{12}, for M_1 against M_2 based on the data Y is given by

$$B_{12} = p(Y|M_1)/p(Y|M_2),$$

where $p(Y|M_k)$ is the integrated likelihood of model M_k $(k = 1, 2)$, namely

$$p(Y|M_k) = \int p(Y|\theta_k, M_k)p(\theta_k|M_k)d\theta_k. \qquad (2.3)$$

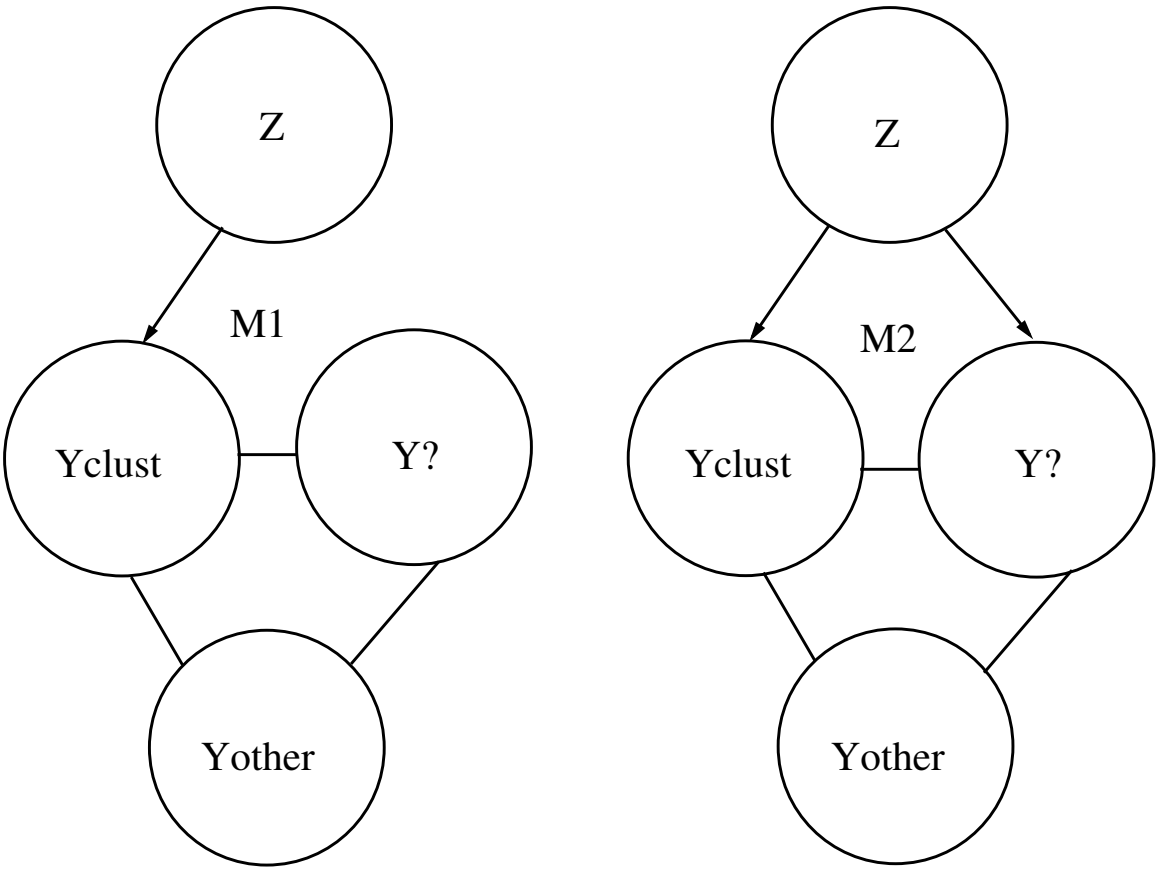

Figure 2.1: Graphical Representation of Models M_1 and M_2 for Clustering Variable Selection. In model M_1, the candidate set of additional clustering variables, $Y^{(?)}$, is conditionally independent of the cluster memberships, $\mathbf{z}$, given the variables $Y^{(clust)}$ already in the model. In model M_2, this is not the case. In both models, the set of other variables considered, $Y^{(other)}$, is conditionally independent of cluster membership given $Y^{(clust)}$ and $Y^{(?)}$, but may be associated with $Y^{(clust)}$ and $Y^{(?)}$.

In (2.3), θ_k is the vector-valued parameter of model M_k, and $p(\theta_k|M_k)$ is its prior distribution [41].

Let us now consider the integrated likelihood of model M_1, $p(Y|M_1) = p(Y^{(clust)}, Y^{(?)}, Y^{(other)}|M_1)$. From (2.2), the model M_1 is specified by three probability distributions: the finite mixture model that specifies $p(Y^{(clust)}|\theta_1, M_1)$, and the conditional distributions $p(Y^{(?)}|Y^{(clust)}, \theta_1, M_1)$ and $p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, \theta_1, M_1)$. We denote the parameter vectors that specify these three probability distributions by θ_{11}, θ_{12}, and θ_{13}, and we assume that their prior distributions are independent. It follows that the integrated likelihood itself factors:

$$p(Y|M_1) = p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_1)\, p(Y^{(?)}|Y^{(clust)}, M_1)\, p(Y^{(clust)}|M_1), \qquad (2.4)$$

where $p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_1) = \int p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, \theta_{13}, M_1)\, p(\theta_{13}|M_1) d\theta_{13}$, and similarly for $p(Y^{(?)}|Y^{(clust)}, M_1)$ and $p(Y^{(clust)}|M_1)$. Similarly, we obtain

$$p(Y|M_2) = p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_2)\, p(Y^{(?)}, Y^{(clust)}|M_2), \qquad (2.5)$$

where $p(Y^{(?)}, Y^{(clust)}|M_2)$ is the integrated likelihood for the mixture model clustering for $(Y^{(?)}, Y^{(clust)})$ jointly.

The prior distribution of the parameter, θ_{13}, is assumed to be the same under M_1 as under M_2. It follows that $p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_2) = p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_1)$. We thus have

$$B_{12} = \frac{p(Y^{(?)}|Y^{(clust)}, M_1) p(Y^{(clust)}|M_1)}{p(Y^{(?)}, Y^{(clust)}|M_2)}, \qquad (2.6)$$

which has been greatly simplified by the cancellation of the factors involving the potentially high-dimensional $Y^{(other)}$. The integrated likelihoods in (2.6) are hard to evaluate analytically, and so we approximate them using the BIC approximation of (2.1).

2.2.3 Greedy Search Algorithm for Clustering Variable Selection

Here we propose a greedy search algorithm. At each stage it searches for the variable to add that most improves the clustering as measured by BIC, and then assesses whether one of the current clustering variables can be dropped. At each stage, the best combination of number of groups and clustering model is chosen. The algorithm stops when no local improvement is possible.

Here is a summary of the algorithm:

1. Select the first clustering variable to be the one which has the most evidence of univariate clustering.

2. Select the second clustering variable to be the one which shows most evidence of bivariate clustering including the first variable selected.

3. Propose the next clustering variable to be the one that shows the most evidence of multivariate clustering including the previous variables selected. Accept this variable as a clustering variable if the evidence favors this over its not being a clustering variable.

4. Propose the variable for removal from the current set of selected clustering variables to be the one for which the evidence of multivariate clustering including all variables selected versus multivariate clustering only on the other variables selected and not on the proposed variable is weakest. Remove this variable from the set of clustering variables if the evidence for clustering is weaker than that for not clustering.

5. Iterate steps 3 and 4 until two consecutive steps have been rejected, then stop.

A more detailed description of this algorithm is given in the appendix to this chapter.

2.2.4 Variable Selection for Model-Based Clustering

We now consider in more detail the case where the mixture components have multivariate normal distributions, that is $f(\cdot|\phi_g) = MVN(\cdot|\mu_g, \Sigma_g)$. This specific form of mixture model clustering is known as model-based clustering For variable selection in this setting, we consider only the case where $Y^{(?)}$ contains just one variable, in which case $p(Y^{(?)}|Y^{(clust)}, M_1)$ represents a normal linear regression model with an intercept and main effects only. This follows from the standard result for conditional multivariate normal means. The BIC approximation to this term in (2.6) is

$$\begin{aligned} 2\log p(Y^{(?)}|Y^{(clust)}, M_1) &\approx BIC_{\text{reg}} \\ &= -n\log(2\pi) - n\log(\text{RSS}/n) \\ &\quad -n - (\dim(Y^{(clust)}) + 2)\log(n), \end{aligned} \tag{2.7}$$

where RSS is the residual sum of squares in the regression of $Y^{(?)}$ on the variables in $Y^{(clust)}$ and $\dim(Y^{(clust)})$ is the number of variables in the set of currently selected clustering variables.

One practical issue with multivariate normal modeling of the components is that, if the model is unconstrained, the number of parameters grows rapidly with the dimension and with the number of clusters, leading to possible overfitting and degradation of performance. For instance, our first example in Section 2.4 below is fairly small, with 4 groups and 5 variables, but the full multivariate normal mixture model still has 83 parameters.

One way of alleviating this is to impose restrictions on the covariance matrices. The covariance matrix can be decomposed, as in [5] and [11], as follows:

$$\Sigma_g = \lambda_g D_g A_g D_g,$$

where λ_g is the largest eigenvalue of Σ_g and controls the volume of the g^{th} cluster, D_g is the matrix of eigenvectors of Σ_g, which control the orientation of that cluster, and A_g

is a diagonal matrix with the scaled eigenvalues as entries, which control the shape of that cluster. By imposing constraints on the various elements of this decomposition, a large range of models is available, ranging from the simple spherical models which have fixed shape, to the least parsimonious model where all elements of the decomposition are allowed to vary across all clusters. A list of the models available in the `mclust` software [31], which allows this type of eigenvalue decomposition Gaussian clustering, is given in Table 2.1. We can choose the parametric model by recognizing that each different combination of number of groups and parametric constraints defines a model, which can then be compared to others using BIC.

Table 2.1: Parameterizations of the Covariance Matrix Σ_g Currently Available in the `mclust` Software. When the data are of dimension 1, only two models are available: equal variances (E), and unequal variances (V).

Name	Model	Distribution	Volume	Shape	Orientation
EII	λI	Spherical	equal	equal	NA
VII	$\lambda_g I$	Spherical	variable	equal	NA
EEI	λA	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_g A$	Diagonal	variable	equal	coordinate axes
EVI	λA_g	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_g A_g$	Diagonal	variable	variable	coordinate axes
EEE	$\lambda D A D^T$	Ellipsoidal	equal	equal	equal
VVV	$\lambda_g D_g A_g D_g^T$	Ellipsoidal	variable	variable	variable
EEV	$\lambda D_g A D_g^T$	Ellipsoidal	equal	equal	variable
VEV	$\lambda_g D_g A D_g^T$	Ellipsoidal	variable	equal	variable

Different choices of subsets of clustering variables also require different covariance structures for the subpopulations. In our examples, we used the `mclust` software, but the method could also be implemented using other mixture modeling software.

Hierarchical agglomerative model-based clustering was used to give the starting values needed in the EM algorithm used to estimate model parameters.

2.3 Simulation Examples

We now present results for two simulation examples. Here we use the term "groups" to refer to the true unknown partition, and we use the term "clusters" to refer to the partition estimated by the clustering algorithm.

2.3.1 Two Groups with Independent Irrelevant Variables

In this simulation there are 150 data points on 7 variables. The data are simulated from a mixture of two multivariate normal distributions with unconstrained (VVV) covariance matrices, so that there are two groups in the data. Only the first two variables contain clustering information. The remaining 5 variables are irrelevant variables independent of the clustering variables, so that the distribution of these variables is multivariate normal independent of group membership. The pairs plot of all the variables is given in Figure 2.2, where X1 and X2 are the clustering variables and X3 to X7 are the independent irrelevant variables.

When forced to cluster on all 7 variables, a five-cluster diagonal EEI model yields the highest BIC value. The model yielding the next highest BIC value is a 4-cluster EEI model. The two-cluster model with the highest BIC value is the two-cluster EEE model but there is a substantial difference of 20 points between this and the model with the highest BIC. This would lead to the (incorrect) choice of a five group structure for this data.

The step by step progress of the greedy search selection procedure is shown in Table 2.2. Two variables are chosen, X1 and X2; these are the correct clustering variables. The two-cluster VVV model has the highest BIC for clustering on these variables by a decisive margin; this gives both the correct number of groups and the correct clustering model (two VVV clusters).

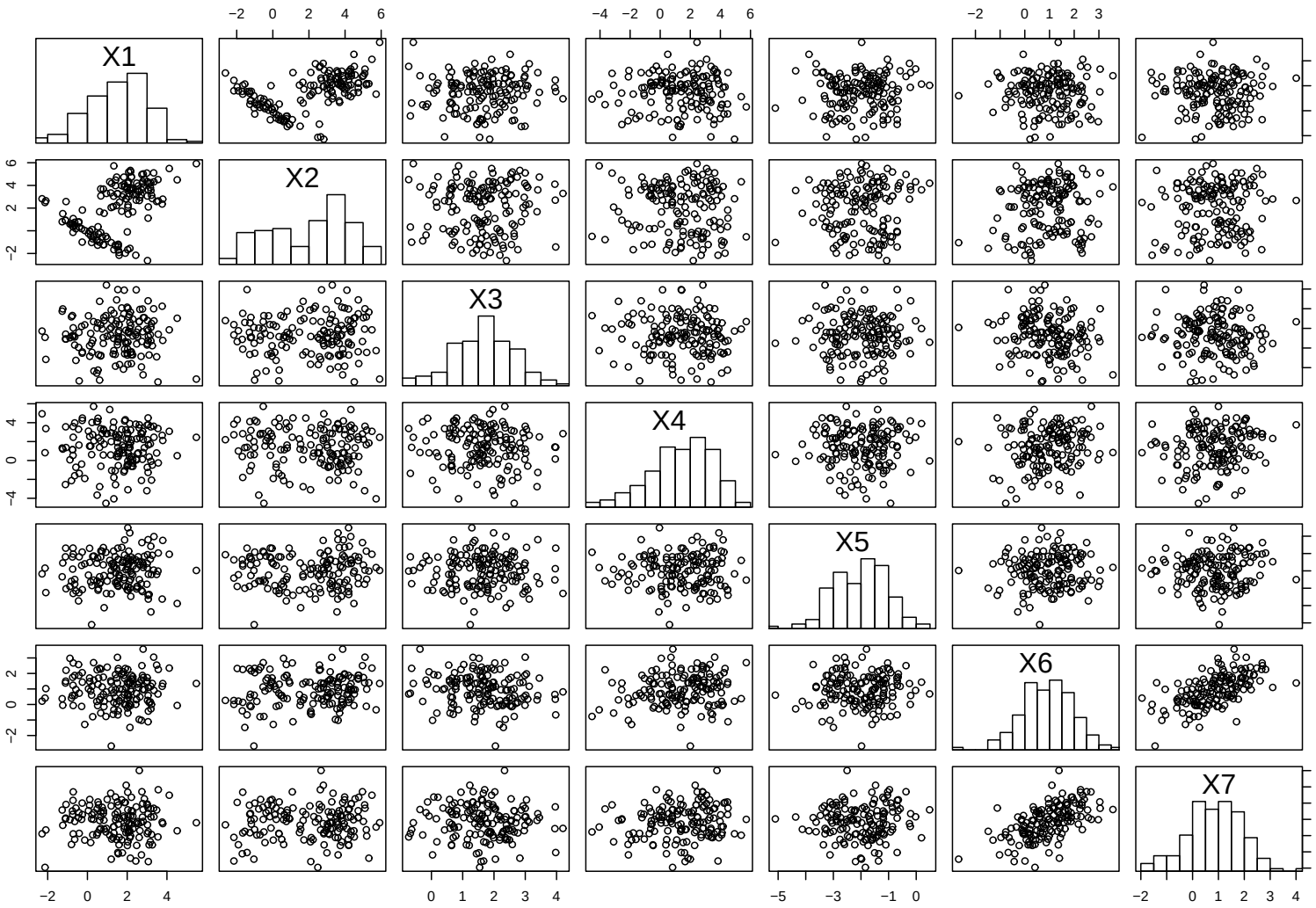

Figure 2.2: First Simulation Example: Pairs plot of the data

Table 2.2: Progress of the Greedy Search Algorithm for the First Simulation Example. The BIC difference is the difference between the BIC for clustering and the BIC for not clustering for the best variable proposed, as given in (3.14) in the Appendix.

Step no.	Best variable proposed	Proposed for	BIC difference	Model chosen	Number of clusters chosen	Result
1	X2	inclusion	15	V	2	Included
2	X1	inclusion	136	VVV	2	Included
3	X6	inclusion	-13	VVV	2	Not included
4	X1	exclusion	136	VVV	2	Not excluded

Since the data are simulated, we know the underlying group memberships of the observations, and we can check the quality of the clustering in this way. The partition arising from clustering on the selected two variables gives 100% correct classification. The confusion matrix for the clustering on all variables is as follows:

	$Group1$	$Group2$
$Cluster1$	53	0
$Cluster2$	4	30
$Cluster3$	34	0
$Cluster4$	1	13
$Cluster5$	0	15

The error rate is 44.7%. This is calculated by taking the best matches of clusters with the groups (i.e. Group 1 $\leftrightarrow$ Cluster 1 and Group 2 $\leftrightarrow$ Cluster 2), which gives us the minimum error rate over all matches between clusters and groups. If we were to correctly amalgamate clusters 1 and 3 and identify them as one cluster, and to amalgamate clusters 2, 4 and 5 and identify them as another cluster, we would get an error rate of 3.3%. However, this assumes knowledge that the investigator would

not typically have in practice.

Finally we pretend (as do many heuristic clustering algorithms) that we know the number of groups (2) correctly in advance, and cluster on all the variables allowing only two-cluster models. The two-cluster model with the highest BIC is the EEE model, and this has an error rate of 3.3%.

In this example, variable selection led to a clustering method that gave the correct number of groups and a 0% error rate. Using all variables led to a considerable overestimation of the number of groups, and a large error rate. Even when the five clusters found in this way were optimally combined into two clusters (with their own mixtures), or when the correct number of groups was assumed known, clustering using all the variables led to a nonzero error rate, with 5 errors.

Table 2.3: Classification Results for the First Simulation Example. The correct number of groups was 2. (c) indicates that the solution was constrained to this number of clusters.

Variable Selection Procedure	Number of variables	Number of clusters	Error rate (%)
None-All variables	7	5	44.7
None-All variables	7	2(c)	3.3
Greedy search	2	2	0

2.3.2 Two Groups with Irrelevant Variables Correlated with Clustering Variables

Again we have 150 points from two clustering variables, with two (VVV) groups. To make the problem more difficult we allow different types of irrelevant variables. There are three independent normal irrelevant variables, seven irrelevant variables which are allowed to be dependent on other irrelevant variables (multivariate normal), and

three irrelevant variables which have a linear relationship with either or both of the clustering variables. This gives a total of 15 irrelevant variables.

The pairs plot of a selection of the variables is given in Figure 2.3. Variables X1 and X2 are the clustering variables, X3 is an independent irrelevant variable, X6 and X7 are irrelevant variables that are correlated with one another, X13 is linearly dependent on the clustering variable X1, X14 is linearly dependent on the clustering variable X2, and X15 is linearly dependent on both clustering variables, X1 and X2.

When forced to cluster on all 15 variables, a two-cluster diagonal EEI model yields the highest BIC. The model yielding the next highest BIC value is a three-cluster diagonal EEI model, with a difference of 10 points between the two. In this case the investigator would probably decide on the correct number of groups, based on this evidence. The error rate for classification based on this model is 1.3%.

The results of the steps when the greedy search selection procedure is run are given in Table 2.4. Two variables are selected, and these are precisely the correct clustering variables. The model with the highest BIC for clustering on these variables is a two-cluster VVV model, and the next highest model in terms of BIC is the three-cluster VVV model. There is a difference of 27 between the two BIC values, which would typically be regarded as strong evidence.

We compare the clustering memberships with the underlying group memberships and find that clustering on the selected variables gives a 100% correct classification, i.e. no errors. In contrast, using all 15 variables gives a nonzero error rate, with two errors. Variable selection has the added advantage in this example that it makes the results easy to visualize, as only two variables are involved after variable selection.

2.4 Real Data Examples

We now give the results of applying our variable selection method to three real datasets in which the "correct" number of groups is known.

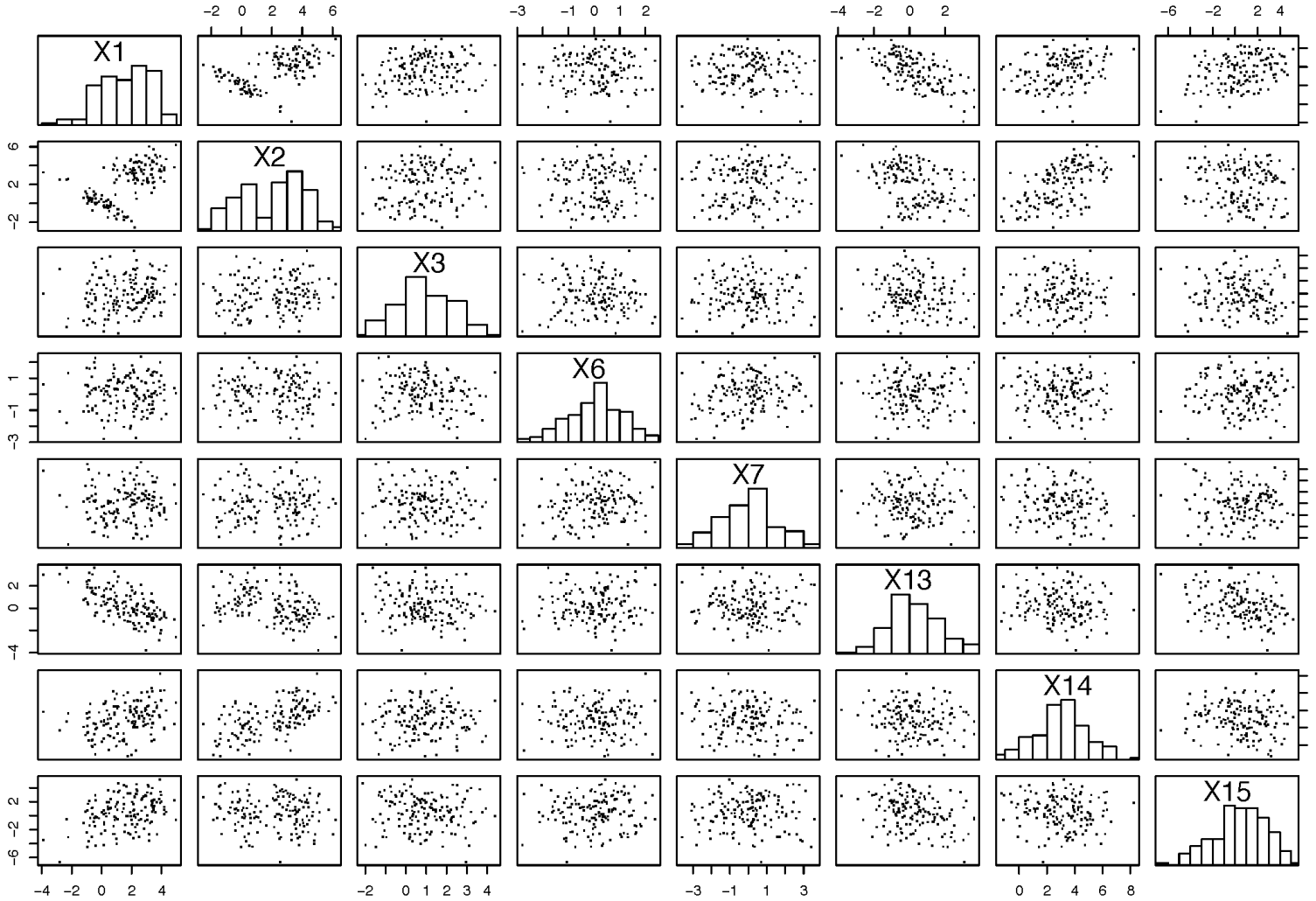

Figure 2.3: Second Simulation Example: Pairs plot of 8 of the 15 variables.

Table 2.4: Progress of the Greedy Search Algorithm for the Second Simulation Example. The BIC difference is the difference between the BIC for clustering and the BIC for not clustering for the best variable proposed, as given in (3.14) in the Appendix.

Step no.	Best variable proposed	Proposed for	BIC difference	Model chosen	Number of clusters chosen	Result
1	X11	inclusion	17	V	2	Included
2	X2	inclusion	5	EEE	2	Included
3	X1	inclusion	109	VVV	2	Included
4	X11	exclusion	-19	VVV	2	Excluded
5	X4	inclusion	-9	VVV	2	Not included
6	X2	exclusion	153	VVV	2	Not excluded

Table 2.5: Classification results for the Second Simulation Example

Variable Selection Procedure	Number of variables	Number of Clusters	Error rate (%)
None-All variables	15	2	1.3
Greedy search	2	2	0

2.4.1 Leptograpsus Crabs Data

This dataset consists of 200 subjects: 100 of species orange (50 male and 50 female), and 100 of species blue (50 male and 50 female). This gives a possible 4 group classification so we are hoping to find a four-cluster structure. There are five measurements on each subject: width of frontal lip (FL), rear width (RW), length along the mid-line of the carapace (CL), maximum width of the carapace (CW) and body depth (BD) in mm. The dataset was published by [10], and was further analyzed by [61] and [52, 53].

The variables selected by the variable selection procedure were (in order of selection) CW, RW, FL and BD. The error rates for the different clusterings are given in Table 2.6. The error rates for the seven-cluster models were the minimum error rates over all matchings between clusters and groups, where each group was matched with a unique cluster.

When no variable selection was done, the number of groups was substantially overestimated, and the error rate was 42.5%, as can be seen in the confusion matrix for clustering on all variables:

	$Group1$	$Group2$	$Group3$	$Group4$
$Cluster1$	32	0	0	0
$Cluster2$	0	31	0	0
$Cluster3$	0	0	28	0
$Cluster4$	0	0	0	24
$Cluster5$	0	0	0	21
$Cluster6$	18	19	0	0
$Cluster7$	0	0	22	5

When our variable selection method was used, the correct number of groups was selected, and the error rate was much lower (7.5%), as can be seen in the confusion

Table 2.6: Classification Results for the Crabs Data. The correct number of groups is 4. (c) indicates that the number of clusters was constrained to this value in advance. The error rates for the 7-cluster models were calculated by optimally matching clusters to groups.

Original Variables				
Variable Selection Procedure	Number of variables	Number of clusters	Model selected	Error rate (%)
None-All variables	5	7	EEE	42.5
None-All variables	5	4(c)	EEE	7.5
Greedy search	4	4	EEV	7.5
Principal Components				
Variable Selection Procedure	Number of components	Number of clusters	Model selected	Error rate (%)
None-All components	5	7	EEE	42.5
None-All components	5	4(c)	EEV	9.0
Greedy search	3	4	EEV	6.5

matrix for the clustering on the selected variables:

	$Group1$	$Group2$	$Group3$	$Group4$
$Cluster1$	40	0	0	0
$Cluster2$	10	50	0	0
$Cluster3$	0	0	50	5
$Cluster4$	0	0	0	45

Variable selection reduced the number of classification errors to a striking extent, especially given that the method selected four of the five variables, so not much variable selection was actually done in this case. This example suggests that the presence of only a single noise variable even in a low-dimensional setting can cause the clustering results to deteriorate.

In clustering, it is common practice to work with principal components of the data, and to select the first several, as a way of reducing the data dimension. Our method could be used as a way of choosing the principal components to be used, and it has the advantage that one does not have to use the principal components that explain the most variation, but can automatically select the principal components that are most useful for clustering. To illustrate this, we computed the five principal components of the data and used these instead of the variables. The variable selection procedure chose (in order) principal components 3, 2 and 1.

Once again, when all the principal components were used, the number of groups was overestimated, and the error rate was high, at 42.5%. When variable selection was carried out, our method selected the correct number of groups without invoking any prior knowledge of the number of groups, and the error rate was much reduced, at 6.5%. Even when the number of groups was assumed to be correctly known in advance, but no variable selection was done, the error rate was higher than with variable selection, at 9.0%.

[13] showed that the practice of reducing the data to the principal components that account for the most variability before clustering is not justified in general. Chang

showed that the principal components with the largest eigenvalues do not necessarily contain the most information about the cluster structure, and that taking a subset of principal components can lead to a major loss of information about the groups in the data. Chang demonstrated this theoretically, by simulations, and in applications to real data. Similar results have been found by other researchers, including [37] for market segmentation, and [72] for clustering gene expression data. Our method to some extent rescues the principal component dimension reduction approach, as it allows one to use all or many of the principal components, and then for clustering select only those that are most useful for clustering, not those that account for the most variance. This avoids Chang's criticism.

In this example, the EM algorithm used for estimating the parameters when clustering on all variables was sensitive to starting values and the best starting values came from randomly generating posterior probabilities rather than hierarchical agglomerative model-based clustering. The variable selection EM clustering was not as sensitive to the starting values, and hierarchical clustering was used to initialize the EM algorithm in that case.

2.4.2 Iris Data

The well-known iris data consist of four measurements on 150 samples of either iris setosa, iris versicolor or iris virginica [1, 29]. The measurements are sepal length, sepal width, petal length and petal width (cm). When one clusters using all the variables, the model with the highest BIC is the two-cluster VEV model, with the three-group VEV model within one BIC point of it. Thus an analyst might conclude that the these data do not contain enough information to decide whether there are two or three groups. The two-group clustering puts versicolor and virginica together, and they are known to be very closely related; their identification as separate species is based in part on information not in this dataset [2]. If one does select the two-group

clustering model slightly favored by BIC, the confusion matrix is as follows:

	$Setosa$	$Versicolor$	$Virginica$
$Cluster1$	50	0	0
$Cluster2$	0	50	50

The setosa group is well picked out but versicolor and virginica have been amalgamated. This leads to a minimum error rate of 33.3%.

The confusion matrix from the three-group clustering is as follows:

	$Setosa$	$Versicolor$	$Virginica$
$Cluster1$	50	0	0
$Cluster2$	0	45	0
$Cluster3$	0	5	50

This gives a 3.3% error rate and reasonable separation. However, given the BIC values, an investigator with no reason to do otherwise might well have erroneously chosen the two-cluster model.

The variable selection procedure selects three variables (all but sepal length). The highest BIC model is the three-cluster VEV model, with the next highest model being the four-cluster VEV model; the BIC difference is 14. The confusion matrix from the three-cluster clustering on these variables is as follows:

	$Setosa$	$Versicolor$	$Virginica$
$Cluster1$	50	0	0
$Cluster2$	0	44	0
$Cluster3$	0	6	50

which is a 4% error rate. A summary of the results from the different methods is given in Table 2.7. For these data, clustering on all variables gives an ambiguous result, while the correct number of groups is decisively chosen when variable selection is done.

Table 2.7: Classification Results for the Iris Data. The correct number of groups is 3. (c) indicates that the number of groups was constrained to this value in advance.

Variable Selection Procedure	Number of variables	Number of Groups	Error rate (%)
None-All variables	4	2	33.3
None-All variables	4	3(c)	3.3
Greedy search	3	3	4

2.4.3 Texture Dataset

The Texture dataset was produced by the Laboratory of Image Processing and Pattern Recognition (INPG-LTIRF) in the development of the Esprit project ELENA No. 6891 and the Esprit working group ATHOS No. 6620. The original source was [7]. This dataset consists of 5500 observations with 40 variables, created by characterizing each pattern using estimation of fourth order modified moments, in four orientations: 0, 45, 90 and 135 degrees; see [38] for details. There are eleven classes of types of texture: grass lawn, pressed calf leather, handmade paper, raffia looped to a high pile, cotton canvas, pigskin, beach sand, another type of beach sand, oriental straw cloth, another type of oriental straw cloth, and oriental grass fiber cloth (labeled groups 1 to 11 respectively). We have 500 observations in each group.

When we clustered on all available variables we found that the model with the highest BIC value was the one-cluster model (with an error rate of 90.9%). When we used the greedy search procedure with a maximum number of 15 clusters, allowing only the unconstrained VVV model since the search space was already so large, we selected 35 variables (all but variables 1, 11, 15, 31 and 40) which, when clustered allowing all models, chose (via BIC) the 11-cluster VVV model. The classification results are shown in Table 2.8.

The classification matrix for the model based on the selected variables is given in Table 2.9. The classification from this model is much closer to the true partition than that from the model based on all the variables, in terms of both the number of groups being correct and the group memberships. We can see that most groups except Group 8, Group 1 and Group 6 are picked out well.

Table 2.8: Classification Results for the Texture Data. The correct number of groups is 11. (c) indicates that the number of clusters was constrained to this value in advance.

Variable Selection Procedure	Number of variables	Number of Clusters	Error rate (%)
None-All variables	40	1	90.9
None-All variables	40	11(c)	40.7
Greedy search	35	11	13.6

2.5 Discussion

We have proposed a method for variable or feature selection in model-based clustering. The method recasts the variable selection problem as one of model choice for the entire dataset, and addresses it using approximate Bayes factors and a greedy search algorithm. For several simulated and real data examples, the method gives better estimates of the number of clusters, lower classification error rates, more parsimonious clustering models, and hence easier interpretation and visualization than clustering using all the available variables.

Our method for searching for the best subset of variables is a greedy search algorithm, and of course this will find only a local optimum in the space of models. The method works well in our experiments, but it may be possible to improve its performance by using a different optimization algorithm, such as Markov chain Monte Carlo

Table 2.9: Texture Data: Confusion matrix for the clustering based on the selected variables. The largest count in each row is boxed.

	Gp 4	Gp 5	Gp 9	Gp 10	Gp 3	Gp 11	Gp 7	Gp 2	Gp 8	Gp 1	Gp 6
Cl 8	**500**	0	0	0	0	0	0	0	0	0	0
Cl 9	0	**500**	0	0	0	0	0	0	0	0	0
Cl 10	0	0	**500**	0	0	0	0	0	0	0	0
Cl 11	0	0	0	**500**	0	1	0	0	0	0	0
Cl 6	0	0	0	0	**499**	0	0	0	0	0	0
Cl 4	0	0	0	0	0	**497**	0	0	0	0	0
Cl 2	0	0	0	0	0	0	**474**	0	0	200	0
Cl 3	0	0	0	0	0	0	0	**439**	0	0	0
Cl 5	0	0	0	0	0	0	0	0	**383**	0	341
Cl 1	0	0	0	0	1	0	0	61	0	**300**	0
Cl 7	0	0	0	0	0	2	26	0	117	0	**159**

or simulated annealing. Our method is analogous to stepwise regression, and this has been found to be often unstable, as noted by [55], for example. This was not a problem for the analyses conducted in this paper however, but it remains an issue to be aware of. Also when the number of variables is vast, for example in microarray data analysis when thousands of genes may be the variables being used, the method is too slow to be practical as it stands. Combining our basic approach with pre-screening (where subsets of variables are selected prior to using the variable selection clustering procedure) and alternative model search methods such as [4]'s headlong procedure (a variant of which is discussed in Section 3.2.3) could yield a method that would be feasible for such cases.

The method is feasible for quite large datasets. For example, when the method was run on a simulated dataset with two clusters, 10,000 observations and 10 variables (of which 8 were clustering variables), using hierarchical clustering on a subset of 1000 observations, a maximum allowed number of 9 clusters and the VVV model only, the CPU time on a laptop with 512 MB of memory and a 1.5 GHz processor was just under 11 hours.

Less work has been done on variable selection for clustering than for classification (or discrimination or supervised learning), reflecting the fact that it is a harder problem. In particular, variable selection and dimension reduction in the context of model-based cluster- ing have not received much attention. One approach that is similar in principle to ours is that given by Dy and Brodley [27] where the feature subset selection is wrapped around EM clustering with order identification. However, they do not consider an eigenvalue de- composition formulation, or both forward and backward steps in their search pattern and there is no explicit model for comparing different feature sets. In a model-based clustering setting Law, Jain, and Figueiredo [45] looked at a wrapper method of feature selection incorporated into the mixture-model formulation. In the first approach each variable is al- lowed to be independent of the others given the cluster membership (diagonal model in the Gaussian setting)

and irrelevant variables are assumed to have the same distribution regard- less of cluster membership. The missing data structure of the EM algorithm is used both to estimate the cluster parameters and to select variables.

Vaithyanathan and Dom [69] put forward an approach which determines both the relevant variables and the number of clusters by using an objective function that incorpo- rates both. The functions used in their paper were integrated likelihood and cross-validated likelihood. The example given was a multinomial model and no extension for continuous or ordinal data was suggested.

Liu, Zhang, Palumbo, and Lawrence [49] proposed a Bayesian approach using MCMC, in which a principal components analysis or correspondence analysis is carried out first and a number of components to be examined are selected. Then the components important for clustering are selected from this subset and clustering is performed simultaneously. The pro- cedure can also automatically select an appropriate Box-Cox transformation to improve the normality of the groups. This approach requires that principal components be used where, in certain cases, investigators may be as interested in the variables important for clustering as in the clustering itself and this information is not easily available in this approach. Also the approach assumes the number of clusters/groups to be known.

An entirely different approach is taken by Lazzeroni and Owen [47], where a two-sided (both variables and samples) cluster analysis is performed which has variable selection as an implicit part of the procedure. Variables are allowed to belong to more than one cluster or to no cluster, and similarly with samples. This was motivated by the analysis of gene expression data. Along a similar line, Getz, Levine, and Domany [34] proposed a method that clusters both variables and samples so that clustering on the subsets found in one will produce stable, sensible clusters in the other. The procedure is iterative but no details on the stopping criterion were given.

McLachlan, Bean, and Peel [54] proposed a dimension reduction method where a mixture of factor analyzers is used to reduce the extremely high dimensionality

of a gene expression problem. Pre-specification of the number of factor analyzers to be used is required. Other examples of dimension reduction include work by Ding, He, Zha, and Simon [24] where cluster membership is used as a bridge between reduced dimensional clusters and the full dimensional clusters and reduces dimensions to one less than the number of clusters. It is an iterative process, swapping between reduced dimensions and the original space. This work focuses mainly on the simplest model, spherical Gaussian clustering. Another dimension reduction technique is given by Chakrabarti and Mehrotra [12], which uses local rather than global correlations. There are a number of parameters, such as the maximum dimension allowed in a cluster, that must be specified, for which the optimal values are not all obvious from the data.

A different approach taken in Mitra, Murthy, and Pal [56], is more similar to a filter selection technique than the wrapper techniques more usually looked at. Since it is a one- step pre-clustering process with no search involved it is very fast, but it takes no account of any clustering structure when selecting the variables. In a similar vein Talavera [66] uses a filter method of subset selection but has no explicit method of deciding how many variables should be used.

Several approaches to variable selection for heuristic clustering methods have been pro- posed. One of the methods of feature selection for the more heuristic distance-based clus- tering algorithms is given by McCallum, Nigam, and Ungar [51] which involves switching between cheap and expensive metrics. A method for k-means clustering variable se- lection is given by Brusco and Cradit [9] which is based on the adjusted RAND index in order to measure similarity of clusterings produced by different variables. However this requires prior specification of number of clusters and there are problems when the variables are highly correlated and there are outliers present in the data. Other methods for variable selection for heuristic clustering include that of Devaney and Ram [23], who consider a stepwise selection search run with the COBWEB hierarchical clustering algorithm.

Friedman and Meulman [32] approach the problem in terms of maximizing an appro- priate function in terms of weights of variables and different clusterings. Different weights are selected depending on the scale of the data for that variable. Since the variables are weighted, rather than selected or removed, there is no actual dimension reduction although it does allow emphasis on different variables for different clusters. The number of groups must be specified by user. Work in a similar vein was done by Gnanadesikan, Kettenring, and Tsao [35]. A similar idea in terms of weighting variables but with a different function to be optimized is suggested by Desarbo, Carroll, Clarck, and Green [21], where the sum of weighted squared distances between data points in groups of variables and a distance based on linear regression on cluster membership is used as the function.

The examples in this chapter have involved continuous data modeled by mixtures of normal distributions. However, the same basic ideas can be applied to variable selection in other clustering contexts, such as clustering multivariate discrete data using latent class models [18, 6] as discussed in chapter 3, or more generally, Bayesian graphical models with a hidden categorical node [15]. When the present approach is adapted to these other clustering problems, it should retain the aspects that make it flexible, especially its ability to simultaneously estimate the number of clusters and group structure, as well as selecting the clustering variables.

Appendix: Greedy Search Variable Selection for Model-Based Clustering Algorithm

Here we give a more complete description of the greedy search variable selection and clustering algorithm for the case of continuous data modeled by multivariate normal groups. This version allows for choosing the number of clusters and the model parameterizations as well, if required, otherwise one can simply alter the steps below slightly to choose only the number of clusters.

- Choose G_{max}, the maximum number of clusters to be considered for the data.

- **First step :** The first clustering variable is chosen to be the one which gives the greatest difference between the BIC of clustering on it (maximized over number of clusters from 2 up to G_{max} and different parameterizations) and BIC of no clustering (single group structure maximized over different parameterizations) on it, where each variable is looked at separately. We do not require that the greatest difference be positive at this point because in certain cases there is no evidence of univariate clustering in data where multivariate clustering may be present and in order to find this clustering we need a starting variable.

  Specifically, we split $Y^{(other)} = Y$ into its variables $y^1, \ldots, y^{D_1}$. For all j in $1, \ldots, D_1$ we compute the approximation to the Bayes factor in (2.6) by

  $$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}}(y^j) - BIC_{\text{not clust}}(y^j)$$

  where $BIC_{\text{clust}}(y^j) = \max_{2 \le G \le G_{max}, m \in \{E,V\}} \{BIC_{G,m}(y^j)\}$, with $BIC_{G,m}(y^j)$ being the BIC given in (2.1) for the model-based clustering model for y^j with G clusters and model m being either the one-dimensional equal-variance (E) or unequal variance model (V), and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}}$ as given in (2.7) (for a regression model with constant mean) with $\dim(Y^{(clust)})$=0.

We choose the best variable, y^{j_1}, such that

$$j_1 = \arg\max_{j:y^j \in Y}(BIC_{\text{diff}}(y^j))$$

and create

$$\begin{aligned} Y^{(clust)} &= (y^{j_1}) \\ \text{and } Y^{(other)} &= Y \backslash y^{j_1} \end{aligned}$$

where $Y \backslash y^{j_1}$ denotes the set of variables in Y excluding variable y^{j_1}.

- **Second step :** Next the set of clustering variables is chosen to be the pair of variables, including the variable selected in the first step, that gives the greatest difference between the BIC for clustering on both variables (maximized over number of clusters from 2 up to G_{max} and different parameterizations) and the sum of the BIC for the univariate clustering of the variable chosen in the first step and the BIC for the linear regression of the new variable on the variable chosen in the first step. Note that we do not assume that the greatest difference is positive since the only criterion the first two variables need to satisfy is being the best initialization variables.

  Specifically, we split $Y^{(other)}$ into its variables $y^1, \ldots, y^{D_2}$. For all j in $1, \ldots, D_2$ we compute the approximation to the Bayes factor in (2.6) by

  $$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}}(y^j) - BIC_{\text{not clust}}(y^j)$$

  where $BIC_{\text{clust}}(y^j) = \max_{2 \leq G \leq G_{max}, m \in M}\{BIC_{G,m}(Y^{(clust)}, y^j)\}$ with $BIC_{G,m}(Y^{(clust)}, y^j)$ being the BIC given in (2.1) for the model-based clustering model for the dataset including both the previously selected variable (contained in $Y^{(clust)}$) and the new variable y^j with G clusters and model m in the set of all possible models M, and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}} + BIC_{\text{clust}}(Y^{(clust)})$ where BIC_{reg} is given in (2.7) (the regression model with independent variable $Y^{(clust)}$ and dependent variable y^j) when $\dim(Y^{(clust)})$=1 (the number of variables currently selected) and

$BIC_{\text{clust}}(Y^{(clust)})$ is the BIC for the clustering with only the currently selected variable in $Y^{(clust)}$.

We choose the best variable, y^{j_2}, with

$$j_2 = \arg \max_{j: y^j \in Y^{(other)}} (BIC_{\text{diff}}(y^j))$$

and create

$$\begin{aligned} Y^{(clust)} &= Y^{(clust)} \cup y^{j_2} \\ \text{and } Y^{(other)} &= Y^{(other)} \backslash y^{j_2} \end{aligned}$$

where $Y^{(clust)} \cup y^{j_2}$ denotes the set of variables including those in $Y^{(clust)}$ and variable y^{j_2}.

- **General Step [Inclusion part] :** The proposed new clustering variable is chosen to be the variable which gives the greatest difference between the BIC for clustering with this variable included in the set of currently selected clustering variables (maximized over numbers of clusters from 2 up to G_{max} and different parameterizations) and the sum of the BIC for the clustering with only the currently selected clustering variables and the BIC for the linear regression of the new variable on the currently selected clustering variables.

- If this difference is positive the proposed variable is added to the set of selected clustering variables. If not the set remains the same.

  Specifically, at step t we split $Y^{(other)}$ into its variables $y^1, \ldots, y^{D_t}$. For all j in $1, \ldots, D_t$ we compute the approximation to the Bayes factor in (2.6) by

$$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}}(y^j) - BIC_{\text{not clust}}(y^j) \tag{2.8}$$

  where $BIC_{\text{clust}}(y^j) = \max_{2 \le G \le G_{max}, m \in M} \{BIC_{G,m}(Y^{(clust)}, y^j)\}$, with $BIC_{G,m}(Y^{(clust)}, y^j)$ being the BIC given in (2.1) for the model-based clustering model for the dataset

including both the previously selected variables (contained in $Y^{(clust)}$) and the new variable y^j with G clusters and model m in the set of all possible models M, and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}} + BIC_{\text{clust}}(Y^{(clust)})$ where BIC_{reg} is given in (2.7) (the regression model with independent variables $Y^{(clust)}$ and dependent variable y^j) when $\dim(Y^{(clust)})$= (the number of variables currently selected) and $BIC_{\text{clust}}(Y^{(clust)})$ is the BIC for the clustering with only the currently selected variables in $Y^{(clust)}$.

We choose the best variable, y^{j_t}, with

$$j_t = \arg \max_{j: y^j \in Y^{(other)}} (BIC_{\text{diff}}(y^j))$$

and create

$$\begin{aligned} Y^{(clust)} &= Y^{(clust)} \cup y^{j_t} \text{ if } BIC_{\text{diff}}(y^{j_t}) > 0 \\ \text{and } Y^{(other)} &= Y^{(other)} \backslash y^{j_t} \text{ if } BIC_{\text{diff}}(y^{j_t}) > 0 \end{aligned}$$

otherwise $Y^{(clust)} = Y^{(clust)}$ and $Y^{(other)} = Y^{(other)}$.

- **General Step [Removal part] :** The proposed variable for removal from the set of currently selected clustering variables is chosen to be the variable from this set which gives the smallest difference between the BIC for clustering with all currently selected clustering variables (maximized over number of clusters greater than 2 up to G_{max} and different parameterizations) and the sum of the BIC for clustering with all currently selected clustering variables except for the proposed variable and the BIC for the linear regression of the proposed variable on the other clustering variables.

- If this difference is negative the proposed variable is removed from the set of selected clustering variables. If not the set remains the same.

  In terms of equations for step $t+1$, we split $Y^{(clust)}$ into its variables $y^1, \ldots, y^{D_{t+1}}$. For all j in $1, \ldots, D_{t+1}$ we compute the approximation to the Bayes factor in

(2.6) by

$$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}} - BIC_{\text{not clust}}(y^j)$$

where $BIC_{\text{clust}} = \max_{2\leq G\leq G_{max}, m\in M}\{BIC_{G,m}(Y^{(clust)})\}$ with $BIC_{G,m}(Y^{(clust)})$ being the BIC given in (2.1) for the model-based clustering model for the dataset including the previously selected variables (contained in $Y^{(clust)}$) with G clusters and model m in the set of all possible models M, and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}} + BIC_{\text{clust}}(Y^{(clust)}\backslash y^j)$ where BIC_{reg} is given in (2.7) (the regression model with independent variables being all of $Y^{(clust)}$ *except* y^j and dependent variable y^j) when $\dim(Y^{(clust)})$= (the number of variables currently selected)-1 and $BIC_{\text{clust}}(Y^{(clust)}\backslash y^j)$ is the BIC for the clustering with all the currently selected variables in $Y^{(clust)}$ except for y^j.

We choose the best variable, $y^{j_{t+1}}$, with

$$j_{t+1} = \arg\min_{j: y^j \in Y^{(clust)}}(BIC_{\text{diff}}(y^j))$$

and create

$$\begin{aligned} Y^{(clust)} &= Y^{(clust)}\backslash y^{j_{t+1}} \text{ if } BIC_{\text{diff}}(y^{j_{t+1}}) \leq 0 \\ \text{and } Y^{(other)} &= Y^{(other)} \cup y^{j_{t+1}} \text{ if } BIC_{\text{diff}}(y^{j_{t+1}}) \leq 0 \end{aligned}$$

otherwise $Y^{(clust)} = Y^{(clust)}$ and $Y^{(other)} = Y^{(other)}$.

- After the first and second steps the general step is iterated until consecutive inclusion and removal proposals are rejected. At this point the algorithm stops as any further proposals will be the same ones already rejected.

Chapter 3

LATENT CLASS ANALYSIS VARIABLE SELECTION

3.1 Introduction

In situations such as surveys, where there are a large number of categorical variables and the question of underlying groupings is of interest, latent class analysis is a parsimonious mixture model method of discovering clusters in a statistically principled way. Because of the underlying statistical model it is possible to use model choice techniques to select the number of clusters/classes believed to be present in the data. However the question of selecting which variables are most important for clustering is one which is not addressed by the latent class model. This question can be important to researchers for substantive reasons and also because restrictions on the number of observations may make it impossible to use all variables to fit a latent class model or to fit a latent class model beyond a certain limited number of classes.

In this chapter an adapted version of the previous chapter's iterative procedure for variable selection using model choice criteria is proposed. This method again uses two models for examining the usefulness of a single variable for clustering, given the variables already selected as being useful for clustering. We then present a new earch algorithm to explore the space of possible models (both in terms of variables selected and number of classes in the model).

In section 3.2.1 we will discuss the general latent class model for categorical data, a necessary condition for identifiability of a latent class model, the rephrasing of comparing different numbers of clusters as a model comparison problem and the Bayesian Information Criterion used to approximate the Bayes factor for answering this problem. In section 3.2.2 we present the two models used to compare the usefulness of a

single variable for clustering versus not clustering, given the variables already selected as useful clustering variables. In section 3.2.3 the headlong search algorithm proposed for exploring the model space is described. Results from simulated examples are given in section 3.3 and results from real data are given in section 3.4. Conclusions and discussion of this approach are given in section 3.5. Finally, the search algorithm from section 3.2.3 is described in greater detail in appendix A and a version of the search algorithm incorporating more efficient latent class model starting value generation is described in appendix B.

3.2 Methodology

3.2.1 Latent Class Analysis

Latent class analysis was first proposed by [46] and although it came before mixture model clustering it still is a specific example in the more general context of this clustering framework. As such, we will first review the idea of mixture model clustering before discussing the particular case of latent class analysis.

We assume that each observation comes from one of a set number of sub-populations in the overall population. The basis of mixture model clustering ([71]) is the idea of modeling each group/sub-population/class with its own density. The overall population is then modeled by a weighted sum of the inidividual sub-populations' densities. The weights are called the mixing distribution or mixture proportions and they represent the proportions of the overall population falling in each sub-population. The weights can also be thought of as the prior probabilities of an observation coming from particular sub-populations. Assuming that the density for group g is f_g we can write the general overall density as:

$$x \sim \sum_{g=1}^{G} \pi_g f_g(x) \tag{3.1}$$

where G is the number of groups, $0 < \pi_g < 1$ and $\sum_{g=1}^{G} \pi_g = 1$.

Often, in practice, the f_g are from the same parametric family (as is the case in latent class analysis) and we can write the general overall density as:

$$x \sim \sum_{g=1}^{G} \pi_g f(x \mid \theta_g)$$

where θ_g is the set of parameters for the g^{th} group.

In general, when looking at discrete data sets in the context of unsupervised learning some simplifying assumptions are usually made in order to avoid over-fitting or indeed inability to fit models to the given data. Latent class analysis is a type of such a parsimonious mixture model clustering for discrete data. The assumption which simplifies the model from the general case of modeling the variables jointly for each group is that of *local independence*. By local independence we mean that conditional on knowing the group an observation came from, the variables are assumed to be independent. This conditional independence can, of course, not be proven in practice but the rationalization underlying latent class analysis is finding the model with the minimum number of classes that will explain the dependence in the data. Each variable, within each group is then modeled with a multinomial density. The general density of a single variable x given it is in group g is then:

$$x \mid g \sim \prod_{j=1}^{d} p_{jg}^{1\{x=j\}} \tag{3.2}$$

where $1\{x = j\}$ is the indicator function equal to 1 if the observation of the variable takes value j and 0 otherwise, p_{jg} is the probability of the variable taking value j in group g and d is the number of possible values or categories the variable can take.

Since we are assuming conditional independence, if we have k variables, their joint group density can be written as a product of their individual group densities. If we have $x = (x_1, \ldots, x_k)$, we can write the general joint group density as:

$$x \mid g \sim \prod_{i=1}^{k} \prod_{j=1}^{d_i} p_{ijg}^{1\{x_i=j\}} \tag{3.3}$$

where $1\{x_i = j\}$ is the indicator function equal to 1 if the observation of the i^{th} variable takes value j and 0 otherwise, p_{ijg} is the probability of variable i taking value j in group g and d_i is the number of possible values or categories the i^{th} variable can take.

The overall density is then a weighted sum of these individual product densities given as:

$$x \sim \sum_{g=1}^{G} (\pi_g \prod_{i=1}^{k} \prod_{j=1}^{d_i} p_{ijg}^{1\{x_i=j\}}) \tag{3.4}$$

where $0 < \pi_g < 1$ and $\sum_{g=1}^{G} \pi_g = 1$.

The model parameters $\{p_{ijg}, \pi_g; i = 1, \ldots, k, j = 1, \ldots, d_i, g = 1, \ldots, G\}$ can be estimated from the data (for a fixed value of G) using the EM algorithm or Newton Rhapson algorithm or a hybrid of the two. These algorithms require starting values which are usually randomly generated. Because the algorithms are not guaranteed to find a global maximum and are usually fairly dependent on good starting values it is routine to generate a number of random starting values and use the best solution given by one of these. In appendix B, an adjusted method useful for the cases where an inordinately large number of starting values is needed to get good estimates of the latent class models and $G > 2$ is presented.

Identifiability

[36] discussed the issue of checking whether a latent class model with a certain number of classes was identifiable for a given number of variables. The necessary condition given for k variables with number of categories $d = (d_1, \ldots, d_k)$ for G classes is

$$\prod_{i=1}^{k} d_i > (\sum_{i=1}^{k} d_i - k + 1) \times G \tag{3.5}$$

which is basically checking that there are enough pieces of information (or cell counts or pattern combinations) to estimate the number of parameters in the model. However, in practice, not all possible pattern combinations are observed (some/many cell

counts may be zero) and so the actual information available may be lower. A more sensible check seems to be

$$\#\text{ of non-zero cell counts} > (\sum_{i=1}^{k} d_i - k + 1) \times G \tag{3.6}$$

When we speak of selecting the number of latent classes in the data we are only considering the numbers for which this minimum criterion is satisfied.

Selecting the number of latent classes in the data

Each different value of G, the total number of latent classes, defines a different model for the data. A method is needed to select the number of latent classes present in the data. Since a statistical model for the data is used, model selection techniques can be applied to this question.

Different numbers of latent classes define different models for the data. In order to choose the best number of classes for the data we need to choose the best model (and the related number of classes). Bayes factors ([42]) are used to compare these models.

The Bayes factor for comparing model M_i versus model M_j is equal to the ratio of the posterior odds for M_i versus M_j to the prior odds for M_i versus M_j. This reduces to the ratio of posterior odds when the prior model probabilities are equal. The general form for the Bayes factor is:

$$B_{ij} = \frac{p(Y \mid M_i)}{p(Y \mid M_j)} \tag{3.7}$$

where $p(Y \mid M_i)$ is known as the integrated likelihood of model M_i (given data Y). $p(Y \mid M_i)$ is called the integrated likelihood because it is obtained by integrating over all of the model parameters (in the latent class analysis case, the mixture proportions and group variable probabilities). Unfortunately the integrated likelihood and thus the ratio is difficult to compute (it has no closed form) and some form of approximation is needed for calculating Bayes factors in practice. We use the Bayesian information criterion (BIC) in our approximation which is very simple to compute.

Bayesian Information Criterion

The Bayesian information criterion (BIC) is defined by

$$\begin{aligned} BIC &= 2 \times \log(\text{maximized likelihood}) \\ &- (\text{no. of parameters}) \times \log(n), \end{aligned} \tag{3.8}$$

where n is the number of observations.

Twice the logarithm of the Bayes factor is approximately equal to the difference between BIC values for the two models being compared. We choose the number of latent classes as discussed before by recognizing that each different number of classes defines a model, which can then be compared to others using BIC. [44] showed BIC to be consistent for the choice of the number of clusters in the context of a restricted form of model-based clustering for normal clusters (where all variables are relevant to the clustering). A rule of thumb for differences in BIC values is a difference of less than 2 is looked at as not really worth mentioning in general, while a difference greater than 10 is seen as constituting strong evidence ([42]).

3.2.2 Variable Selection Model

At any stage in the procedure we can partition the collection of variables into three sets: $Y^{(clust)}, Y^{(?)} and Y^{(other)}$, where:

- $Y^{(clust)}$ is the set of variables already selected as useful for clustering,
- $Y^{(?)}$ is the variable(s) being considered for inclusion into/exclusion from Y^{clust},
- $Y^{(other)}$ is the set of all other variables.

Given this partition and the (unknown) clustering memberships z we can recast the question of $Y^{(?)}$'s usefulness for clustering as a model selection question using two

different models; M_1 which assumes $Y^{(?)}$ is not useful for clustering and M_2 which assumes $Y^{(?)}$ is useful for clustering.

$$\begin{aligned} M_1 : p(Y|\mathbf{z}) &= p(Y^{(clust)}, Y^{(?)}, Y^{(other)}|\mathbf{z}) \\ &= p(Y^{(other)}|Y^{(?)}, Y^{(clust)})p(Y^{(?)})p(Y^{(clust)}|\mathbf{z}) \qquad (3.9) \\ M_2 : p(Y|\mathbf{z}) &= p(Y^{(clust)}, Y^{(?)}, Y^{(other)}|\mathbf{z}) \\ &= p(Y^{(other)}|Y^{(?)}, Y^{(clust)})p(Y^{(?)}, Y^{(clust)}|\mathbf{z}), \\ &= p(Y^{(other)}|Y^{(?)}, Y^{(clust)})p(Y^{(?)}|\mathbf{z})p(Y^{(clust)}|\mathbf{z}), \end{aligned}$$

where $\mathbf{z}$ is the (unobserved) set of cluster memberships. Model M_1 specifies that, given $Y^{(clust)}$, $Y^{(?)}$ is independent of the cluster memberships (defined by the unobserved variables $\mathbf{z}$), that is, $Y^{(?)}$ gives no information about the clustering. Model M_2 implies that $Y^{(?)}$ does provide information about clustering membership, beyond than given just by $Y^{(clust)}$. This follows the approach used in the previous chapter for model-based clustering with continuous data and normal clusters with the difference that conditional independence of the variables was not assumed there and instead of $p(Y^{(?)})$ in model M_1 we had $p(Y^{(?)} \mid Y^{(clust)})$ which assumed conditional independence instead of full independence, i.e. the assumption in model M_1 previously was that given the information in $Y^{(clust)}$, $Y^{(?)}$ had no *additional* clustering information. The difference between the assumptions underlying the two models is illustrated in Figure **??**, where arrows indicate dependency.

We assume that the remaining variables $Y^{(other)}$ are conditionally independent of the clustering given $Y^{(clust)}$ and $Y^{(?)}$ and belong to the same parametric family in both models.

Models M_1 and M_2 are compared via an approximation to the Bayes factor which allows the high-dimensional $p(Y^{(other)}|Y^{(clust)}, Y^{(?)})$ to cancel from the ratio. The Bayes factor, B_{12}, for M_1 against M_2 based on the data Y is given by

$$B_{12} = p(Y|M_1)/p(Y|M_2),$$

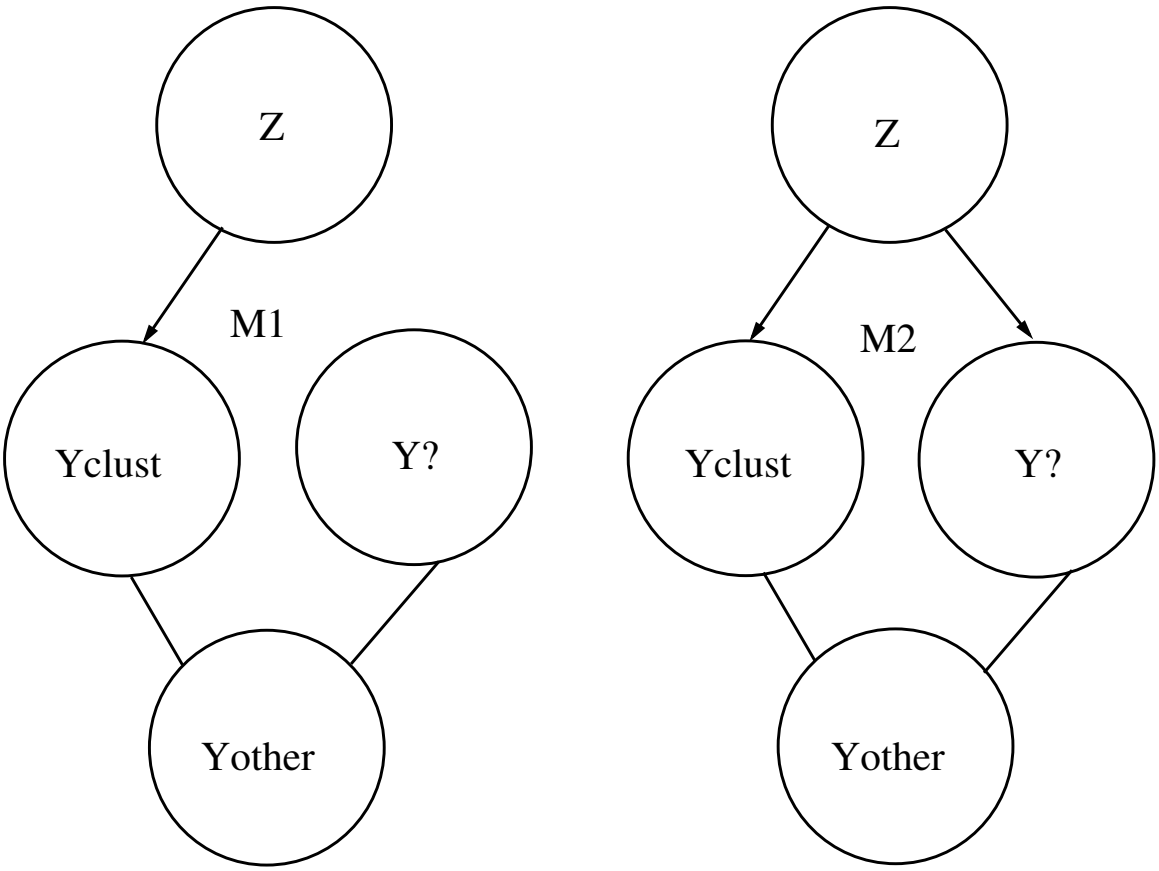

Figure 3.1: Graphical Representation of Models M_1 and M_2 for Latent Class Variable Selection. In model M_1, the candidate set of additional clustering variables, $Y^{(?)}$, is independent of the cluster memberships, $\mathbf{z}$, given the variables $Y^{(clust)}$ already in the model. In model M_2, this is not the case. In both models, the set of other variables considered, $Y^{(other)}$, is conditionally independent of cluster membership given $Y^{(clust)}$ and $Y^{(?)}$, but may be associated with $Y^{(clust)}$ and $Y^{(?)}$.

where $p(Y|M_k)$ is the integrated likelihood of model M_k $(k = 1, 2)$, namely

$$p(Y|M_k) = \int p(Y|\theta_k, M_k)p(\theta_k|M_k)d\theta_k. \tag{3.10}$$

In (3.10), θ_k is the vector-valued parameter of model M_k, and $p(\theta_k|M_k)$ is its prior distribution ([42]).

Let us now consider the integrated likelihood of model M_1, $p(Y|M_1) = p(Y^{(clust)}, Y^{(?)}, Y^{(other)}|M_1)$. From (3.9), the model M_1 is specified by three probability distributions: the latent class model that specifies $p(Y^{(clust)}|\theta_1, M_1)$, and the distributions $p(Y^{(?)}|\theta_1, M_1)$ and $p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, \theta_1, M_1)$. We denote the parameter vectors that specify these three probability distributions by θ_{11}, θ_{12}, and θ_{13}, and we assume that their prior distributions are independent. It follows that the integrated likelihood itself factors:

$$p(Y|M_1) = p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_1)\, p(Y^{(?)}|M_1)\, p(Y^{(clust)}|M_1), \tag{3.11}$$

where $p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_1) = \int p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, \theta_{13}, M_1)\, p(\theta_{13}|M_1)d\theta_{13}$, and similarly for $p(Y^{(?)}|M_1)$ and $p(Y^{(clust)}|M_1)$. Similarly, we obtain

$$p(Y|M_2) = p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_2)\, p(Y^{(?)}, Y^{(clust)}|M_2), \tag{3.12}$$

where $p(Y^{(?)}, Y^{(clust)}|M_2)$ is the integrated likelihood for the latent class model for $(Y^{(?)}, Y^{(clust)})$.

The prior distribution of the parameter, θ_{13}, is assumed to be the same under M_1 as under M_2. It follows that $p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_2) = p(Y^{(other)}|Y^{(?)}, Y^{(clust)}, M_1)$. We thus have

$$B_{12} = \frac{p(Y^{(?)}|M_1)p(Y^{(clust)}|M_1)}{p(Y^{(?)}, Y^{(clust)}|M_2)}, \tag{3.13}$$

which has been greatly simplified by the cancellation of the factors involving the potentially high-dimensional $Y^{(other)}$. The integrated likelihoods in (3.13) are hard to evaluate analytically, and so we approximate them using the BIC approximation of (3.8).

3.2.3 Headlong Search Algorithm

Given these models we need to find a method for creating partitions at each step. Initially we need enough variables to start $Y^{(clust)}$ so that a latent class model for $G > 1$ can be identified. If a latent class model on the set of all variables is identifiable for $G > 1$ then we choose the best latent class model that can be identified and we can rank the variables according to the sum of the variability of their categories' probabilities across the groups, with the assumption that greater variability implies greater separation of the classes on these variables implying greater importance for these variables in the clustering. Given this ranking we choose the top minimum number of variables that allow a latent class model with $G > 1$ to be identified. This is our starting $Y^{(clust)}$. The other variables can be left in their ordering based on variability for future order of introduction in the stepwise algorithm.

In the case where the above strategy is not possible, a number of alternatives can be used. The minimum number of variables needed for identification of a latent class model with $G > 1$ can be calculated and a selection of random subsets of this number of variables can be chosen and the variable set which gives the greatest overall average variability of categories' probabilities across the groups (given the best latent class model identified) is chosen for the initial $Y^{(clust)}$. In small cases it may be possible to enumerate all possible subsets to choose the best initial $Y^{(clust)}$.

Once we have an initial set of clustering variables $Y^{(clust)}$ we can proceed with the general inclusion and exclusion steps of the headlong algorithm.

First we must define constants *upper* and *lower*, where *upper* is the quantity above which the difference in BIC for models M_2 and M_1 will result in a variable being included in $Y^{(clust)}$ and below which the difference in BIC for models M_2 and M_1 will result in a variable being excluded from $Y^{(clust)}$, and *lower* is the quantity below which the difference in BIC for models M_2 and M_1 will result in a variable being removed from consideration for the rest of the procedure. A natural value for *upper*

is 0, by which we mean that any positive difference in BIC for models M_2 and M_1 is taken as evidence of a variable's usefulness for clustering and any negative difference is taken as evidence of a variable's lack of usefulness, whereas a difference of *lower* is taken to indicate that a variable is unlikely to ever be useful as a clustering variable and is no longer even checked. In general a large negative number such as -100 (which by our rule of thumb would constitute strong evidence against) makes a sensible value for *lower*.

- *Inclusion Step*: Propose each variable in $Y^{(other)}$ singly in turn for $Y^{(?)}$. Calculate the difference in BIC for models M_2 and M_1 given the current $Y^{(clust)}$.
  If the variable's BIC difference is:
  - not above *upper* but above *lower*, do not include in $Y^{(clust)}$ and return variable to the end of the list of variables in $Y^{(other)}$
  - below *upper* and *lower*, do not include in $Y^{(clust)}$ and remove variable from $Y^{(other)}$
  - above *upper*, include variable in $Y^{(clust)}$ and stop inclusion step

  If we reach the end of the list of variables in $Y^{(other)}$ the inclusion step is stopped

- *Exclusion Step*: Propose each variable in $Y^{(clust)}$ singly in turn for $Y^{(?)}$ (with the remaining variables in $Y^{(clust)}$ not including current $Y^{(?)}$ now defined as $Y^{(clust)}$ in M_1 and M_2).Calculate the difference in BIC for models M_2 and M_1.
  If the variable's BIC difference is:
  - below *upper* but above *lower*, exclude the variable from (the original) $Y^{(clust)}$ and return variable to the end of the list of variables in $Y^{(other)}$ and stop exclusion step
  - below *upper* and *lower*, exclude the variable from (the original) $Y^{(clust)}$ and from $Y^{(other)}$ and stop exclusion step

– above *upper*, do not exclude the variable from (the original) $Y^{(clust)}$

If we reach the end of the list of variables in $Y^{(clust)}$ the exclusion step is stopped

If $Y^{(clust)}$ remains the same after consecutive inclusion and exclusion steps the headlong algorithm algorithm is stopped (since the set will not change again).

This search algorithm is different from the greedy search algorithm described in section 2.2.3 in two ways:

1. The best variable (in terms of the BIC difference) is not necessarily selected in each inclusion and exclusion step in the headlong search.

2. It is possible that some variables are not looked at in any step after a certain point in the headlong algorithm (after being removed from consideration).

3.3 Simulated Data Results

3.3.1 Binary data example

Five hundred points are simulated from a two class model satisfying the local independence assumption. There are four variables that separate the classes (variables 1–4) and nine variables that are noise variables, i.e. have the same probabilities in each class (variables 5–13). The true model parameters are reported in Table 3.1.

When we estimate the latent class model based on all thirteen variables BIC selects a 2 class model as being the best fit for the data. Since we have simulated the data and know the true membership of each point we can compare the partition given by the true classification with that produced by the model estimated using all the variables. The number of observations incorrectly classified by this model was 123. The number of observations that would be incorrectly classified by using the model with the true parameters is 110. The estimated parameters from the model with all variables are given in Table 3.2.

Table 3.1: True model parameters for binary data example

Mixture proportions		
	Class 1	Class 2
	0.6	0.4
Variable	Prob. of success in class 1	Prob. of success in class 2
1	0.6	0.2
2	0.8	0.5
3	0.7	0.4
4	0.6	0.9
5	0.5	0.5
6	0.4	0.4
7	0.3	0.3
8	0.2	0.2
9	0.9	0.9
10	0.6	0.6
11	0.7	0.7
12	0.8	0.8
13	0.1	0.1

Table 3.2: Estimated parameters for the model involving all variables for the binary data example

Mixture proportions		
	Class 1	Class 2
	0.56	0.44
Variable	Prob. of success in class 1	Prob. of success in class 2
1	0.60	0.19
2	0.85	0.56
3	0.71	0.35
4	0.61	0.86
5	0.57	0.44
6	0.37	0.45
7	0.35	0.21
8	0.16	0.19
9	0.89	0.93
10	0.59	0.62
11	0.82	0.64
12	0.80	0.80
13	0.06	0.13

The variables ordered according the variability of their estimated probabilities (in decreasing order) are: 1, 3, 2, 4, 11, 7, 5, 6, 13, 9, 8, 10, 12. As expected the first four variables are our clustering variables. We note that the difference between the true probabilities across groups is 0.4 for variable 1 and 0.3 for variables 2 to 4. Since variable 1 therefore gives better separation of the classes, we would expect it to be first in the list.The number of variables needed in order to estimate a latent class model with at least 2 classes is 3. So the set of variables suggested for starting clustering variables is {1, 3, 2}. The individual step results for the variable selection procedure starting with this set are given in Table 3.3.

Table 3.3: Results for each step of the variable selection procedure for the binary data example

Variable(s) Proposed	Step Type	Clustering BIC	# of Classes	Independence BIC	Difference	Accepted?
1, 3, 2	Inclusion	-1976.35	2	-1981.25	4.90	Accepted
4	Inclusion	-2565.37	2	-2573.62	8.25	Accepted
11	Inclusion	-3148.76	2	-3146.72	-2.04	Rejected
4	Exclusion	-2565.37	2	-2573.62	8.25	Rejected

When clustering on the four selected variables only, BIC again chooses 2 classes as the best fitting model. Comparing the partition gotten by classifying observations based on the estimates from this model and the true partition we find that 110 observations have been misclassified which seems to be optimal given that this is the error also gotten from classifying based on the true model parameters. The estimated parameters from the model using only selected variables are given in Table 3.4.

The misclassification results are summarized in Table 3.5.

Table 3.4: Estimated parameters for the model involving only the selected variables for the binary data example

Mixture proportions		
	Class 1	Class 2
	0.64	0.36
Variable	Prob. of success in class 1	Prob. of success in class 2
1	0.56	0.17
2	0.83	0.52
3	0.72	0.26
4	0.63	0.89

Table 3.5: Misclassification Summary for the binary data example. Recall that the number of misclassifications for the model based on the true parameters was 110

Variables Included	No. of obs. misclassified
All	123
1,2,3,4	110

3.3.2 Non binary data example

One thousand points are simulated from a three class model satisfying the local independence assumption. There are four variables that separate the classes (variables 1–4) and six variables that are noise variables, i.e. which have the same probabilities in each class (variables 5–10). The true model parameters are reported in Table 3.6 and Table 3.7.

Table 3.6: True clustering parameters for the model with data from variables with different numbers of categories

Mixture proportions				
		Class 1	Class 2	Class 3
		0.3	0.4	0.3
Variable	Category	Prob. of category in class 1	Prob. of category in class 2	Prob. of category in class 3
Var. 1	Cat. 1	0.1	0.3	0.6
	Cat. 2	0.1	0.5	0.2
	Cat. 3	0.8	0.2	0.2
Var. 2	Cat. 1	0.5	0.1	0.7
	Cat. 2	0.5	0.9	0.3
Var. 3	Cat. 1	0.2	0.7	0.2
	Cat. 2	0.2	0.1	0.6
	Cat. 3	0.3	0.1	0.1
	Cat. 4	0.3	0.1	0.1
Var. 4	Cat .1	0.1	0.6	0.4
	Cat. 2	0.5	0.1	0.4
	Cat. 3	0.4	0.3	0.2

When we estimate the latent class model based on all ten variables BIC selects a 2 (instead of 3) class model as being the best fit for the data. The difference between BIC values for a 2 class and a 3 class model based on all variables is 68. Again, since we have simulated the data and know the true membership of each point we can compare the partition given by the true classification with that produced by the 2-class model estimated using all the variables. A cross-tabulation of the true memberships versus the estimated memberships from the 2 class model with all variables is given below

		Estimated classes	
		1	2
True classes	1	293	25
	2	85	324
	3	245	28

The misclassification rate from using the model with the true parameters is 19.9%. If we match each true class to the best estimated class in the 2 class model with all variables we get a misclassification rate of 38.3%. If we assume that we knew the number of classes in advance to be 3 then the misclassification rate for the 3 class model with all variables is reduced to 25.7%. However this is knowledge that is not typically available in practice.

The variables ordered according the variability of their estimated probabilities in the 2 class model (in decreasing order) are: 2, 3, 1, 4, 6, 9, 7, 10, 8, 5. The first four variables are our clustering variables. The number of variables needed in order to estimate a latent class model with at least 2 classes is 3. So the set of variables suggested for starting clustering variables is {2, 3, 1}. The individual step results for the variable selection procedure starting with this set are given in Table 3.8

When clustering on the four selected variables only, BIC this time chooses 3 classes as the best fitting model. Comparing the partition from classifying observations based on the estimates from this model and the true partition we find that the misclassifi-

Table 3.7: True non-clustering parameters for the model with data from variables with different numbers of categories

Mixture proportions				
		Class 1	Class 2	Class 3
		0.3	0.4	0.3
Variable	Category	Prob. of category in class 1	Prob. of category in class 2	Prob. of category in class 3
Var. 5	Cat. 1	0.4	0.4	0.4
	Cat. 2	0.5	0.5	0.5
	Cat. 3	0.1	0.1	0.1
Var. 6	Cat. 1	0.2	0.2	0.2
	Cat. 2	0.4	0.4	0.4
	Cat. 3	0.1	0.1	0.1
	Cat. 4	0.3	0.3	0.3
Var. 7	Cat. 1	0.2	0.2	0.2
	Cat. 2	0.3	0.3	0.3
	Cat. 3	0.3	0.3	0.3
	Cat. 4	0.1	0.1	0.1
	Cat. 5	0.1	0.1	0.1
Var. 8	Cat. 1	0.2	0.2	0.2
	Cat. 2	0.8	0.8	0.8
Var. 9	Cat. 1	0.7	0.7	0.7
	Cat. 2	0.1	0.1	0.1
	Cat. 3	0.2	0.2	0.2
Var. 10	Cat. 1	0.1	0.1	0.1
	Cat. 2	0.2	0.2	0.2
	Cat. 3	0.1	0.1	0.1
	Cat. 4	0.6	0.6	0.6

Table 3.8: Results for each step of the variable selection procedure for the data from variables with different numbers of categories

Variable(s) Proposed	Step Type	Clustering BIC	# of Classes	Independence BIC	Difference	Accepted?
2, 3, 1	Inclusion	-6122.65	2	-6193.37	70.72	Accepted
4	Inclusion	-8235.05	3	-8330.71	95.66	Accepted
8	Inclusion	-9261.46	3	-9248.28	-13.18	Rejected
2	Exclusion	-8235.05	3	-8322.40	87.36	Rejected

cation rate is 23.8%. The estimated parameters from the model using only selected variables are given in Table 3.9.

The misclassification results are summarized in Table 3.10.

3.4 Real Data Examples

3.4.1 ICU Data

The ICU dataset comes from Appendix 2 of "Applied Logistic Regression ([40], [48]). The dataset consists of observations on 200 different subjects who formed a subset of a larger study on survival rates of adult patients admitted to an intensive care unit at Baystate Medical Center in Springfield, Massachusetts. There were two classes of patients, those (40) who died and those (160) who survived. It is hoped that latent class analysis on the other 16 variables will pick up this underlying structure reasonably well and that the variables found to be important to the clustering can tell us which measures/tests could be more important for prognosis in ICU cases.

Information is available about the patient's gender (male/female) [Gender], race (white/black/other) [Race], service at admission (medical/surgical) [Servad], if cancer

Table 3.9: Estimated parameters for the model involving only the selected variables for the data from variables with different numbers of categories

Mixture proportions				
		Class 1	Class 2	Class 3
		0.40	0.43	0.16
Variable	Category	Prob. of category in class 1	Prob. of category in class 2	Prob. of category in class 3
Var. 1	Cat. 1	0.10	0.34	0.85
	Cat. 2	0.1	0.49	0.13
	Cat. 3	0.80	0.17	0.02
Var. 2	Cat. 1	0.49	0.12	0.82
	Cat. 2	0.51	0.88	0.18
Var. 3	Cat. 1	0.21	0.64	0.17
	Cat. 2	0.27	0.14	0.63
	Cat. 3	25	0.13	0.08
	Cat. 4	0.27	0.09	0.12
Var. 4	Cat .1	0.14	0.53	0.39
	Cat. 2	0.47	0.10	0.47
	Cat. 3	0.39	0.37	0.14

Table 3.10: Misclassification Summary for the data from variables with different numbers of categories. (c) indicates that the number of classes was constrained to this value in advance. Recall that the minimum misclassification rate from the model based on the true parameters is 19.9%.

Variables Included	No. of Classes selected	Misclassification Rate
All	2	38.3%
All	3(c)	25.7%
1,2,3,4	3	23.8%

were part of the problem (yes/no) [Cancer], if there was a history of chronic renal problems (yes/no) [CRN], probable infection (yes/no) [Infect], whether CPR had been performed prior to admission (yes/no) [CPR], whether the patient had been previously admitted to the ICU (yes/no) [Previcu], type of admission (elective/emergency) [Admit], whether there was a fracture (yes/no) [Fract], PO2 level in initial bloodwork ($\geq 60/\leq 60$) [PO2], PH level in initial bloodwork ($7.25/< 7.25$) [PH], PCO2 level in initial bloodwork ($45/> 45$) [PCO2], bicarbonate in initial bloodwork ($18/< 18$) [Bicarb], creatinine in initial bloodwork ($2/> 2$) [Creat] and unconsiousness at ICU (none/stupor/coma) [Consc].

When BIC is used to select the number of classes in a latent class model with all of the variables, it decisively selects 2 (with a difference of at least 30 points between 2 classes and any other identifiable number of classes). When the variables are put in decreasing order of variance of estimated probabilities between classes the ordering is the following: Servad, Admit, Infect, Bicarb, Cancer, PO2, CPR, PH, PCO2, CRN, Gender, Consc, Creat, Fract, Previcu and Race.

Observations were classified into whichever group their estimated membership probability was greatest for. The partition estimated by this method is compared

with the true partition below:

	Lived	Died
Class 1	92	11
Class 2	68	29

If class 1 is matched with the lived class and class 2 with the died class there is a misclassification rate of 39.5%.

The variable selection method chooses 11 variables, all except Fract, Gender, Previcu, Consc and Race. This does not mean these 5 variables are not useful for classifying observations in general but that they do not add any extra class information over the 11 variables selected. BIC again selected 2 classes for the latent class model on the selected variables. Again the estimated partition from this model is compared to the true partition.

	Lived	Died
Class 1	93	12
Class 2	67	28

The misclassification rate is again 39.5%. Now the partition from the model involving all variables is compared to that of the model only using the selected variables:

	Sel. Var. Class 1	Sel. Var. Class 2
All Var. Class 1	103	0
All Var. Class 2	2	95

Clearly the only difference between the partitions is that two observations classified as class 2 in the model with all variables are classified as class 1 in the model with only the selected variables. One error is made in each instance. Apart from these two observations, the largest difference in estimated group membership probabilities between the two latent class models is 0.1. This is unsurprising as the estimated model parameters in the variables common to both latent class models and the mixing proportions differ between models by at most 0.03.

3.4.2 Hungarian Heart Disease Data

This dataset consists of five categorical variables from a larger dataset (with 10 other continuous variables) collected from the Hungarian Institute of Cardiology. Budapest by Andras Janosi, M.D ([22], [33]). The outcome of interest is diagnosis of heart disease into two categories: $< 50\%$ diameter narrowing and $> 50\%$ diameter narrowing. Originally there was information about 294 subjects but 10 subjects had to be removed due to missing data. The five variables given are: gender (male//female) [sex], chest pain type (typical angina/atypical angina/non-anginal pain/asymptomatic) [cp], fasting blood sugar > 120 mg/dl (true/false) [fbs], resting electrocardiographic results (normal/having ST-T wave abnormality/showing probable or definite left ventricular hypertrophy by Estes' criteria) [restecg] and exercise induced angina (yes/no) [exang].

When BIC is used to select the number of classes in a latent class model with all of the variables, it decisively selects 2 (with a difference of at least 38 points between 2 classes and any other identifiable number of classes). When the variables are put in decreasing order of variance of estimated probabilities between classes the ordering is the following: cp, exang, sex, restecg and fbs.

Observations were classified into whichever group their estimated membership probability was greatest for. The partition estimated by this method is compared with the true partition below:

	<50% narrowing	>50% narrowing
Class 1	134	13
Class 2	47	90

If class 1 is match with the <50% class and class 2 with the >50% class there is a misclassification rate of 21.2%.

The variable selection method chooses 3 variables: cp, exang and sex. BIC selects 2 classes for the latent class model on these variables. The partition given by this model is exactly the same as the one given by the model with all variables. The largest

difference in estimated group membership probabilities between the two latent class models is 0.1. The estimated model parameters in the variables common to both latent class models and the mixing proportions differ between models by at most 0.003.

The estimated parameters for the latent class model with all variables included is given in Table 3.11.

3.5 Discussion

As demonstrated by the simulated examples in the previous sections, the variable selection procedure selects the correct variables and using only selected variables in latent class models can help improve both the misclassification rate and the selection of the correct number of underlying classes in the data. The medical examples given show that a smaller subset of measurements can be used to classify the subjects which could help improve the speed of diagnosis/prognosis without degrading the classification performance.

In general it appears to be a better idea to perform some kind of variable selection prior to attempting to estimate the final clustering model either in the discrete data case or the continuous data case. We have seen that inclusion of noise variables can have degrading effects on two important aspects of clustering: model estimation and choice of number of clusters.

In terms of estimation of the model, including variables with no cluster structure can either smear out separated clusters/classes or introduce spurious classes. It is difficult without any extra knowledge to know what can happen in advance. From looking at the simulations and data sets presented here as well as others, it would appear that these problems only occur when separation between the classes is poor in general.

Although [44] showed that BIC was consistent for choice of number of classes in the case of restricted (multivariate) normal and poisson mixture models, this work was

Table 3.11: Estimated parameters for the model involving all variables for Hungarian Heart Disease Data

Mixture proportions			
		Class 1	Class 2
		0.49	0.51
Variable	Category	Prob. of category in class 1	Prob. of category in class 2
Chest Pain Type	Typical angina	0.07	0.00
	Atypical angina	0.64	0.08
	Non-anginal pain	0.29	0.08
	Asymptomatic	0.00	0.83
Exercise Induced Angina	No	0.98	0.42
	Yes	0.02	0.58
Gender	Female	0.38	0.16
	Male	0.62	0.84
Resting Electrocardiographic Results	Normal	0.82	0.80
	Having ST-T wave abnormality	0.15	0.20
	Showing probable or definite left ventricular hypertrophy by Estes' criteria	0.03	0.01
Fasting blood sugar > 120 mg/dl	False	0.94	0.92
	True	0.06	0.08

done assuming that all variables were relevant to the clustering. Empirical evidence seems to suggest that when noise/irrelevant variables are present the result no longer holds true. The general correctness of the BIC approximation in a specific case of binary variables with two classes in a naive bayesian network (which is equivalent to a 2-class latent class model with the local independence assumption satisfied) was looked at in [62]. The authors found that although the traditional BIC penalty term of # of parameters $\times$ log(# of observations) (or half this depending on the definition) was correct for regular points in the data space, it was not correct for singularity points (with two different types of singularity points requiring two adjusted versions of the penalty term). Similarly in the case of redundant or irrelevant variables being included (which is closely related to the two singularity point types) they found that the two adjusted penalty terms were correct.

These issues with clustering with noise make it imperative for some form of variable selection to be done in order for appropriate models to be found.

Appendix A: Headlong Search Variable Selection for Latent Class Clustering Algorithm

Here we give a more complete description of the headlong search variable selection and clustering algorithm for the case of discrete data modeled by conditionally independent multinomially distributed groups. This version allows for choosing the number of clusters and the model parameterizations as well, if required, otherwise one can simply alter the steps below slightly to choose only the number of clusters. Note that for each latent class model fitted in this algorithm one must run a number of random starts to find the best estimate of the model (in terms of BIC). We recommend at least 5 for small to medium problems but for bigger problems hundreds may be needed to get a decent model estimate. The issue of getting good starting values without multiple generation of random starts is dealt with in appendix B.

- Choose G_{max}, the maximum number of clusters/classes to be considered for the data. Make sure that this number is identifiable for your data! Define constants *upper* (default 0) and *lower* (default -100), where *upper* is the quantity above which the difference in BIC for models M_2 and M_1 will result in a variable being included in $Y^{(clust)}$ and below which the difference in BIC for models M_2 and M_1 will result in a variable being excluded from $Y^{(clust)}$, and *lower* is the quantity below which the difference in BIC for models M_2 and M_1 will result in a variable being removed from consideration for the rest of the procedure.

- **First step :** One way of choosing the initial clustering variable set is by estimating a latent class model with at least 2 classes for all variables (if more classes are identifiable, estimate all identifiable class numbers and choose the model with the best number of classes via BIC). Order the variables in terms of variability of their estimated probabilities across classes. Choose the minimum top variables that allow at least a 2-class model to be identified. This is the initial $Y^{(clust)}$. We do not require that the BIC difference between clustering

and a model with a single component for our $Y^{(clust)}$ to be positive at this point because we need a set of starting variables for the algorithm. These can be removed later if there are not truly clustering variables.

Specifically we estimate the $\{p_{ijg}, i = 1, \ldots, k, j = 1, \ldots, d_i, g = 1, \ldots, G\}$ where k is the number of variables, d_i is the number of categories for the i^{th} variables and G is the number of classes. For each variable i we calculate $V(i) = \sum_{j=1}^{d_i} Var(p_{ijg})$. We order the variables in decreasing order of $V(i)$: $y^{(1)}, y^{(2)}, \ldots, y^{(k)}$ and find m the minimum number of top variables that will identify a latent class model with $G \geq 2$.

$$\begin{aligned} Y^{(clust)} &= \{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\} \\ Y^{(other)} &= \{y^{(m+1)}, \ldots, y^{(k)}\} \end{aligned}$$

If the previous method is not possible (data cannot identify latent class model for $G > 1$) then split the variables randomly into subsets with enough variables to identify a latent class model for at least 2 classes, estimate the latent models for each subset and calculate the BICs, estimate the single component models for each subset and calculate the 1 class BICs and choose the subset with the highest difference between latent class model and 1 component model BICs as the initial $Y^{(clust)}$.

Specifically look at the list of categories $d = (d_1, \ldots, d_k)$ and work out the minimum number of variables m that allows a latent class model for $G \geq 2$ to be identified. Split the variables into S subsets of at least m variables in each. For each set $Y_s, s = 1, \ldots, S$ estimate:

$$BIC_{\text{diff}}(Y_s) = BIC_{\text{clust}}(Y_s) - BIC_{\text{not clust}}(Y_s)$$

where $BIC_{\text{clust}}(Y_s) = \max_{2 \leq G \leq G_{maxs}} \{BIC_G(Y_s)\}$, with $BIC_G(Y_s)$ being the BIC given in 3.8 for the latent class model for Y_s with G classes and G_{maxs} being the maximum number of identifiable classes for Y_s, and $BIC_{\text{not clust}}(Y_s) = BIC_1(Y_s)$.

We choose the best variable subset, Y_{s^1}, such that

$$s^1 = \arg \max_{s:Y_s \in Y} (BIC_{diff}(Y_s))$$

and create

$$\begin{aligned} Y^{(clust)} &= Y_{s^1} \\ \text{and } Y^{(other)} &= Y \backslash Y_{s^1} \end{aligned}$$

where $Y \backslash Y_{s^1}$ denotes the set of variables Y excluding the subset Y_{s^1}.

- **Second step :** Next we look at each variable in $Y^{(other)}$ singly in order as the new variable under consideration for inclusion into $Y^{(clust)}$. For each variable we look at the difference between the BIC for clustering on the set of variables including the variables selected in the first set and the new variable (maximized over number of clusters from 2 up to G_{max}) and the sum of the BIC for the clustering of the variables chosen in the first step and the BIC for the single class latent class model for the new variable. If this difference is less than *lower* the variable is removed from consideration for the rest of the procedure and we continue checking the next variable. Once the difference is greater than *upper* we stop and this variable is included in the set of clustering variables. Note that if no variable has difference greater than *upper* we include the variable with the largest difference in the set of clustering variables. We force a variable to be selected at this stage to give one final extra starting variable.

  Specifically, we split $Y^{(other)}$ into its variables $y^1, \ldots, y^{D_2}$. For each j in $1, \ldots, D_2$ until $BIC_{\text{clust}}(y^j) > upper$, we compute the approximation to the Bayes factor in (3.13) by

$$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}}(y^j) - BIC_{\text{not clust}}(y^j)$$

  where $BIC_{\text{clust}}(y^j) = \max_{2 \leq G \leq G_{max}j} \{BIC_G(Y^{(clust)}, y^j)\}$ with $BIC_G(Y^{(clust)}, y^j)$ being the BIC given in (3.8) for the latent class clustering model for the dataset

including both the previously selected variables (contained in $Y^{(clust)}$) and the new variable y^j with G classes, and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}} + BIC_{\text{clust}}(Y^{(clust)})$ where BIC_{reg} is $BIC_1(y^j)$ and $BIC_{\text{clust}}(Y^{(clust)})$ is the BIC for the latent class clustering model with only the currently selected variables in $Y^{(clust)}$.

We choose the best variable, y^{j_2}, with

$$j_2 = \arg \max_{j: y^j \in Y^{(other)}} (BIC_{\text{diff}}(y^j))$$

and create

$$\begin{aligned} Y^{(clust)} &= Y^{(clust)} \cup y^{j_2} \\ \text{and } Y^{(other)} &= Y^{(other)} \backslash y^{j_2} \end{aligned}$$

where $Y^{(clust)} \cup y^{j_2}$ denotes the set of variables including those in $Y^{(clust)}$ and variable y^{j_2}.

- **General Step [Inclusion part] :** Each variable in $Y^{(other)}$ is proposed singly (in order), until the difference between the BIC for clustering with this variable included in the set of currently selected clustering variables (maximized over numbers of clusters from 2 up to G_{max}) and the sum of the BIC for the clustering with only the currently selected clustering variables and the BIC for the single class latent class model of the new variable, is greater than *upper*.

- The variable with BIC difference greater than upper is then included in the set of clustering variables and we stop the step. Any variable whose BIC difference is less than *lower* is removed from consiferation for the rest of the procedure. If no variable has BIC difference greater than *upper* no new variable is included in the set of clustering variables

  Specifically, at step t we split $Y^{(other)}$ into its variables $y^1, \ldots, y^{D_t}$. For j in $1, \ldots, D_t$ we compute the approximation to the Bayes factor in (3.13) by

$$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}}(y^j) - BIC_{\text{not clust}}(y^j) \tag{3.14}$$

where $BIC_{\text{clust}}(y^j) = \max_{2 \leq G \leq G_{max}j}\{BIC_G(Y^{(clust)}, y^j)\}$, with $BIC_G(Y^{(clust)}, y^j)$ being the BIC given in (3.8) for the latent class clustering model for the dataset including both the previously selected variables (contained in $Y^{(clust)}$) and the new variable y^j with G clusters, and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}} + BIC_{\text{clust}}(Y^{(clust)})$ where BIC_{reg} is the single class latent class model for variable y^j and $BIC_{\text{clust}}(Y^{(clust)})$ is the BIC for the clustering with only the currently selected variables in $Y^{(clust)}$.

We check if $BIC_{\text{diff}}(y^j) > upper$,
if so we stop and set

$$\begin{aligned} Y^{(clust)} &= Y^{(clust)} \cup y^{j_t} \text{ if } BIC_{\text{diff}}(y^{j_t}) > 0 \\ \text{and } Y^{(other)} &= Y^{(other)} \backslash y^{j_t} \text{ if } BIC_{\text{diff}}(y^{j_t}) > 0 \end{aligned}$$

if not we increment j and re-calculate $BIC_{\text{diff}}(y^j)$ If $BIC_{\text{diff}}(y^j) < lower$ we remove it from both $Y^{(clust)}$ and $Y^{(other)}$

If no j has $BIC_{\text{diff}}(y^j) > upper$ leave $Y^{(clust)} = Y^{(clust)}$ and $Y^{(other)} = Y^{(other)}$.

- **General Step [Removal part] :** Each variable in $Y^{(clust)}$ is proposed singly (in order), until the difference between the BIC for clustering with this variable included in the set of currently selected clustering variables (maximized over numbers of clusters from 2 up to G_{max}) and the sum of the BIC for the clustering with only the other currently selected clustering variables (and not the variable under consideration) and the BIC for the single class latent class model of the variable under consideration, is less than $upper$.

- The variable with BIC difference less than upper is then removed from the set of clustering variables and we stop the step. If the difference is greater than $lower$ we include the variable at the end of the list of variables in $Y^{(other)}$. If not we remove it entirely from consideration for the rest of the procedure.If no variable

has BIC difference less than *upper* no variable is excluded from the current set of clustering variables

In terms of equations for step $t+1$, we split $Y^{(clust)}$ into its variables $y^1, \ldots, y^{D_{t+1}}$. For each j in $1, \ldots, D_{t+1}$ we compute the approximation to the Bayes factor in (3.13) by

$$BIC_{\text{diff}}(y^j) = BIC_{\text{clust}} - BIC_{\text{not clust}}(y^j)$$

where $BIC_{\text{clust}} = \max_{2 \leq G \leq G_{max}} \{BIC_G(Y^{(clust)})\}$ with $BIC_{G,m}(Y^{(clust)})$ being the BIC given in (3.8) for the model-based clustering model for the dataset including the previously selected variables (contained in $Y^{(clust)}$) with G clusters, and $BIC_{\text{not clust}}(y^j) = BIC_{\text{reg}} + BIC_{\text{clust}}(Y^{(clust)} \backslash y^j)$ where BIC_{reg} is the single class latent class model for variable y^j and $BIC_{\text{clust}}(Y^{(clust)} \backslash y^j)$ is the BIC for the clustering with all the currently selected variables in $Y^{(clust)}$ except for y^j.

We check if $BIC_{\text{diff}}(y^j) < upper$,
if so we stop and set

$$\begin{aligned} Y^{(clust)} &= Y^{(clust)} \backslash y^{j_t} \text{ if } BIC_{\text{diff}}(y^{j_t}) < upper \\ \text{and } Y^{(other)} &= Y^{(other)} \cup y^{j_t} \text{ if } lower < BIC_{\text{diff}}(y^{j_t}) < upper \end{aligned}$$

if not we increment j and re-calculate $BIC_{\text{diff}}(y^j)$ If $BIC_{\text{diff}}(y^j) < lower$ we remove it from both $Y^{(clust)}$ and $Y^{(other)}$

If no j has $BIC_{\text{diff}}(y^j) < upper$ leave $Y^{(clust)} = Y^{(clust)}$ and $Y^{(other)} = Y^{(other)}$.

- After the first and second steps the general step is iterated until consecutive inclusion and removal proposals are rejected. At this point the algorithm stops as any further proposals will be the same ones already rejected.

Appendix B: Headlong Search Variable Selection for Latent Class Clustering Algorithm with Smart Starting Values

In the previous appendix we discussed the details of the headlong algorithm for latent class variable selection. In each step multiple latent class models for different set of data/variables and classes are estimated. Previously we have only mentioned that starting values are generated randomly for each model several times and the best (in terms of BIC/likelihood) of the resulting estimated models is chosen as the single estimate for a particular latent class model. This means that for each different dataset and each different number of classes we are required to generate random starting values and estimate the model via EM numerous times. For datasets with reasonable numbers of variables this is not too computationally expensive but for more complex datasets it is burdensome. Also with increasing numbers of observations and/or variables and/or classes more random starts are needed to have any confidence in finding the global maximum likelihood for the model as the likelihood surface becomes more complex, with increasing numbers of local maxima.

Because of the stepwise nature of the algorithm we can use models estimated before to give good starting values for new models. By starting values here we mean the matrix z of conditional probabilities of membership in the different components for each observation.

At the end of each step (either inclusion or exclusion) we have a set of currently selected clustering variables. At some point in the step we have estimated the latent class model for this set over a range of classes (or sometimes just one, 2 classes) and chosen the model with the number of classes that gives us the highest BIC. We can call this model $LCA_{current}$ and the number of classes in this best model for the current set of clustering variables $G_{current}$. We can also save the z matrix for this model and call it $z_{current}$.

In our next step we will be either looking at models for $Y^{(clust)}$ with a new ad-

ditional variable (inclusion step) or models for $Y^{(clust)}$ leaving out one of the current clustering variables. It seems obvious that a reasonable starting z matrix for models involving the new dataset (which is either a sub- or super-set of the old one) and number of classes $G_{current}$ would be $z_{current}$, because the dataset will only have changed by one variable. So instead of randomly generating multiple z matrixes (or other starting parameters) to try to get the global maximum likelihood for our latent class model, we merely use what we believe to be a good set starting z matrix (which hopefully will be reasonably close to the global maximum in the new likelihood space).

However, we may still wish to have good starting values for the new dataset with different numbers of classes, $G_{current} \pm c$. But our $z_{current}$ will be an $n \times G_{current}$ matrix (where n is the number of observations) and we need $n \times (G_{current} \pm c$ matrices. How can we sensibly create a new matrix with c more/less columns given our $z_{current}$?

We will look at the case for $+1$ and -1 separately (the analogue for general $+c$ and $-c$ should be obvious). It will be rare in practice to need more than ± 1 at each step as the number of identifiable classes will only generally increase with the number of variables selected.

For -1 we want to reduce the number of columns of our $z_{current}$ by 1. A sensible way to do this is to collapse the two closest classes (in terms of Euclidean distance in the parameter space). We calculate the distances between the classes' estimated parameters/probabilities from $LCA_{current}$ and select the closest two. We then simply remove the two columns corresponding to those classes from $z_{current}$ and replace them with one column equal to the sum (across rows) of the removed columns. This is our new starting z matrix for the model with $G_{current} - 1$ classes. In terms of a single observation with probability p_1 of being in the first chosen class and probability p_2 of being the second chosen class we are saying the observation has probability $p_1 + p_2$ of being in the new class created from the amalgamation of the two i.e. the observation will be in the new class if he is in either of the old classes. Note that if we wish to, we can weight the distances with the mixing proportions, making it more likely that

we would join smaller close classes.

For $-c$ we can use the resulting matrix from the process described in the previous paragraph to estimate the model for $G_{current} - 1$ classes and then reduce the resulting estimated z from this model by one column in the same fashion, continuing on in the same way until we have removed c columns.

For $+1$ we want to increase the number of columns of our $z_{current}$ by 1. An obvious way to do this is by splitting a class in two. We choose the largest class (in terms of mixing proportions). We then remove the column corresponding to that class from $z_{current}$ and call this w and estimate a two class latent class model using the data points weighted by w. Obviously we have returned to problem of needing starting values for estimating our 2-class model. However usually a small number of randomly generated starts, say 5, for this number of classes will result in an estimated model achieving the global maximum likelihood and this is usually not too computationally expensive. Once we have our 2-class model estimate of the z matrix, called z_2, we can multiply this by w and add the resulting two columns to the original $z_{current}$ (less the removed column), giving us a starting z matrix for estimating the $G_{current} + 1$ class model. We can think of w as being the conditional probability of an observation being in the old selected class and then the new z_2 matrix as being the probability for an observation being in either of the two new sub-classes *given* it was in the old class.

Again for $+c$ we can use the resulting matrix from the process described in the previous paragraph to estimate the model for $G_{current}+1$ classes and then increase the resulting estimated z from this model by one column in the same fashion, continuing on in the same way until we have added c columns.

Chapter 4

NORMAL UNIFORM MIXTURE DIFFERENTIAL GENE EXPRESSION DETECTION FOR CDNA MICROARRAYS

4.1 Introduction

Differentially expressed genes between two or more samples may be of interest to researchers for different reasons, for example, looking at causes of or treatments for diseases such as cancer. Given appropriately processed data, the researcher needs a methodology for assessing the genes in order to separate out ones of interest, i.e. genes with "significantly" different levels of expression in different samples. Widely used methods for single slide data include examining the ratio of expression levels for the gene in each of the two samples/channels (or the log ratio), which was the quantity examined in one of the first statistical analyses for differential expression in cDNA microarrays [14]. One of the earliest uses of this quantity for determining differential expression was the "rule of two", where if the gene's ratio of expression levels in the two channels/samples is greater than two or less than half, it is considered to be differentially expressed [64].

Methods for data with replicate slides include the standard t test, which requires adjustment for the multiple comparisons being made. Modifications of this approach to account for multiple comparisons include the approach of Dudoit, Yang, Callow and Speed [26], which used a permutation analysis on Welsh's t-statistics, and the Significance Analysis of Microarrays (SAM) method, which modifies the t-statistic by adding a constant to the denominator [68]. A good summary of multiple testing adjustments is given by [25].

The idea of modeling the data as two groups of genes, one differentially expressed

and one not, seems to be a natural and intuitive approach. This approach has been used in the context of a Bayesian analysis [58], EBarrays, assuming that the observed ratios had a gamma distribution the reciprocal of whose scale parameter itself had a gamma distribution, or, as an alternative assumption, that the observed log ratios were normally distributed and the prior for the mean was normal also. A two-component mixture model was used to model the two groups and the posterior probabilities were used to make inference about differential expression. This follows from work done for single slide data with a Gamma-Gamma hierarchical model [57]. Another approach using mixture models is given by Pan, Lin and Le [59].

This chapter presents a very simple methodology based on mixture models called Normal Uniform Differential Gene Expression (NUDGE) detection. It is applicable to both single slide and replicated cDNA microarray datasets, produced by two of the more widely used experimental setups. After standardizing, the log ratio (or averaged across replicates log ratio) observations are modeled with a two-component mixture model; a normal component for those genes that are not differentially expressed and a uniform component for those that are. The mixture gives posterior probabilities of differential expression which do not need to be adjusted for multiple testing. This methodology is applied to three different experiments. The experiments include single replicate data (Like-Like), multiple replicate data (HIV and Apo A1), experiments with different samples being labeled with their own dyes (HIV) and experiments with all samples being labeled with one dye and compared to a reference sample (Apo A1). The results given by NUDGE are compared with those given by some other methodologies for these types of cDNA microarray experiments (different comparison methods used for different types of experiments).

4.2 Methods

4.2.1 Model for Detecting Differential Expression

Our methods are applied to normalized average log ratios; we discuss the specification of these quantities in different experimental settings in the section on normalizations below. In this section we will refer to them simply as observed log ratios. We use logarithms to base 2.

Our model is a normal-uniform mixture model [5, 67]. We begin by modeling the genes as two different groups: differentially expressed and non-differentially expressed. Each group is modeled by its own density, and so the data as a whole are modeled by a weighted mixture of these densities, where the weights correspond to the prior probabilities of being in each of the two groups. This results in a mixture model with two components. Since genes that are not differentially expressed have a true log ratio of zero with some measurement/other error, we model the observed log ratios for these genes, after an appropriate transformation, as a group with a Gaussian density. The differentially expressed genes have log ratios that are, for the most part, in some sense "far" from the other group. So these genes can be viewed as outliers from the main distribution of non-differentially expressed genes. These genes are modeled as uniformly distributed over an appropriately wide range.

The model is

$$x_i \overset{\text{iid}}{\sim} \pi N(x_i|\mu,\sigma^2) + (1-\pi)U_{[a,b]}(x_i), \quad i = 1,\ldots,N, \tag{4.1}$$

where x_i is the observed log ratio for gene i, π is the prior probability that a gene is not differentially expressed, $N(x|\mu,\sigma^2)$ denotes a Gaussian distribution with mean μ and variance σ^2, $U_{[a,b]}(x)$ denotes a uniform distribution on the interval $[a,b]$, and N is the number of genes.

We estimate the model by maximum likelihood using the EM algorithm [20]. We define the unknown labels, $z_i,\ i = 1,\ldots,N$, where z_i is 0 if gene i is not differentially

expressed and 1 if it is. There are two steps in the algorithm: the Expectation, or E step, where the labels are estimated given the current parameter estimates, and the Maximization, or M step, where the model parameters, π, μ and σ^2, are estimated given the current estimates of the labels. The maximum likelihood estimates of a and b are $\hat{a} = \min\{x_i : i = 1, \ldots, N\}$, and $\hat{b} = \max\{x_i : i = 1, \ldots, N\}$; these do not change during the algorithm. The steps in the algorithm are as follows:

***Iteration* k**

Expectation Step

- $\hat{z}_i^{(k)} = \frac{(1-\hat{\pi}^{(k-1)})U_{[\hat{a},\hat{b}]}(x_i)}{\hat{\pi}^{(k-1)}N(x_i|\hat{\mu}^{(k-1)},(\hat{\sigma}^{(k-1)})^2)+(1-\hat{\pi}^{(k-1)})U_{[\hat{a},\hat{b}]}(x_i)}, i = 1, \ldots, N.$

Maximization Step

- $\hat{\pi}^{(k)} = \frac{\sum_{i=1}^{N}(1-\hat{z}_i^{(k)})}{N}$,
- $\hat{\mu}^{(k)} = \frac{\sum_{i=1}^{N}(1-\hat{z}_i^{(k)})\times x_i}{\sum_{i=1}^{N}(1-\hat{z}_i^{(k)})}$,
- $(\hat{\sigma}^{(k)})^2 = \frac{\sum_{i=1}^{N}(1-\hat{z}_i^{(k)})\times(x_i-\hat{\mu}^{(k)})^2}{\sum_{i=1}^{N}(1-\hat{z}_i^{(k)})}$.

The likelihood for the model given parameter estimates at iteration k is

$$L(\mathbf{x}; \hat{\pi}^{(k)}, \hat{\mu}^{(k)}, (\hat{\sigma}^{(k)})^2) = \prod_{i=1}^{N}\{\hat{\pi}^{(k)}N(x_i; \hat{\mu}^{(k)}, (\hat{\sigma}^{(k)})^2) + (1 - \hat{\pi}^{(k)})U_{[\hat{a},\hat{b}]}(x_i)\}. \quad (4.2)$$

The above steps are iterated until convergence. Convergence can be checked by calculating the parameter estimates, the labels, and the logarithm of the likelihood at each step, given the current estimates of the parameters. Once the change in these quantities between steps gets small enough, the algorithm is deemed to have converged. The increasing property of the EM algorithm guarantees that a local maximum is reached, but a global maximum cannot be guaranteed. This depends on the starting values. For the analyses in this chapter, the starting value for the label z_i was 1 if gene i's observed log ratio, minus the mean value for all genes and divided by

the standard deviation of the values across all genes, was greater than 2 in absolute value, and 0 otherwise. This appeared to give good results.

The final label estimate for gene i, $\hat{z}_i$, is the posterior probability that it is differentially expressed, given the parameter estimates. The posterior probabilities do not need to be adjusted for multiple comparisons.

4.2.2 Normalizations

There are two different types of experimental setup for which we will discuss normalization. The first is where the two different samples, say control and treatment, have each been labeled with a different color dye, say treatment with red (Cy5, R) and control with green (Cy3, G). In the second experimental setup, the treatment and control samples have replicates, with both control and treatment replicates being labeled with the same dye, say red (Cy5, R), and these are compared to a reference sample labeled with the other dye.

Two of the data sets analyzed in this chapter, the HIV and the Like-like datasets, are of the first type of setup. The other data set analyzed in this chapter is the Apo AI mouse data [26] which is of the second type of setup, with pooled control mRNA used as its reference sample. Since there are slightly different normalizations and quantities of interest used for analysis in these two cases, we will discuss them separately below, referring to the first experiment type as the log ratio experiment (since the log ratios are the quantities of interest), and to the second as the log ratio difference experiment (since the differences of log ratios between control and treatment samples are the quantities of interest).

Normalizations for the log ratio cDNA Experiment

The main problem in applying the Normal-Uniform mixture model is that the data need to be normalized in order for this model to be appropriate. In the basic type of cDNA experiment, the log ratio of expressions in the two samples is the quantity

of interest. There are dye and other effects that add a bias, making the mean of the non-differentially expressed log ratios non-zero (see the Like-like example in the Results section). Also, the variance of the log ratios depends on the log of the total intensity, where the total intensity is defined as the product of the red and green intensities. We also need to ensure that any normalization does not "pull in" the differentially expressed genes.

Single slide normalizations

The normalization of single slide log ratios is a two-step process. In the first step, the observed log ratios are regressed nonparametrically on the log total intensities, using the lowess regression smoother [16], and the fitted value is subtracted from the observed log ratios. In our implementation, a modification, the loess smoother [17], is used in place of lowess. Specifically,

$$\log_{norm}\left(\frac{R}{G}\right) = \log\left(\frac{R}{G}\right) - c(\log(RG)), \tag{4.3}$$

where R and G are the intensities in the red and green channels, and $c(\log(RG))$ is the fitted value from loess regression of $\log(R/G)$ on $\log(RG)$, a situation we denote by $c(\log(RG)) = loess(\log\left(\frac{R}{G}\right) \sim \log(RG))$. We got good results with a loess span in the range 60% to 80%. This generally did a good job of normalizing the mean but not the spread.

The spread depends on the log intensity, $\log(RG)$, and we estimate a running mean absolute deviation by loess regression of the absolute mean-normalized log ratio on the log total intensity. We then divide the mean-normalized log ratio by the loess-estimated mean absolute deviation in order to get our final estimate,

$$\log_{normv}\left(\frac{R}{G}\right) = \frac{\log_{norm}\left(\frac{R}{G}\right)}{c_v(\log\left(RG\right))}, \tag{4.4}$$

where $c_v(\log(RG)) = loess(|\log_{norm}\left(\frac{R}{G}\right)| \sim \log(RG))$. We got good results with a span between 10% and 20%. As can be seen from the figures in the Results section,

this does a good job of making the log ratios for non-differentially expressed genes approximately normal and homoscedastic.

Multiple slide normalizations with dye swap

In dye swap experiments, there is an even number of replicates and they are divided into two groups with equal numbers of replicates. In the second group of replicates, the assignment of dyes to samples is the reverse of that in the first group. Log ratios in this case are taken with the different samples set as the numerator and denominator (since the assigned dyes will be different for the two groups and averaging must be done over the same ratio of samples not the same ratio of dyes). In that case, mean normalization is unnecessary, although normalization of the variance is still required. This is because we take the average of the log ratios across replicates, ensuring that the dye effect cancels out.

Multiple slide normalizations without dye swap

Here we take the average of log ratios and log total intensities across replicates for each of the genes and apply the mean lowess normalization, given by equation 4.3, with average log ratios and total intensities in place of the single replicate log ratios and total intensities.

The variance normalization is not the same for multiple replicate slides as for a single slide. Because the average log ratios are not robust to outliers, even after mean normalization, we carry out a normalization based on variation across replicates rather than on variation depending on intensities, to downweight the influence of outlying observations. If the empirical standard deviation of the log ratios across replicates is greater than the absolute mean-normalized average log ratio for a gene, we divide its mean-normalized average log ratio by its standard deviation. If the empirical standard deviation of the log ratios across replicates is small, defined as smaller than the absolute mean-normalized average log ratio, we divide instead by a constant.

The constant is chosen to be a high percentile (we use the 99th) of the distribution of the standard deviations of genes for which the absolute mean-normalized average log ratio is greater than the standard deviation. This avoids a gene being declared differentially expressed just because its empirical across-replicate standard deviation is small, as can easily happen by chance when there are few replicates.

Thus the mean- and variance-normalized log ratio for a given gene is:

$$\log_{normv}\left(\frac{R}{G}\right) = \frac{\overline{q}}{s}1_{[|\overline{q}|<s]} + \frac{\overline{q}}{k}1_{[|\overline{q}|\geq s]}, \tag{4.5}$$

where $\overline{q}$ is the mean-normalized average log ratio, s is the standard deviation of log ratios across replicates, and k is the chosen percentile of the distribution of standard deviations of genes whose absolute mean-normalized average log ratio is greater than their standard deviation.

Normalizations for the log ratio difference cDNA Experiment

Here the quantity of interest is the difference in average log ratios between control and treatment replicates.

We define

$$M = \frac{1}{n_{treatment}}\sum_{i=1}^{n_{treatment}} q_{treatment,i} - \frac{1}{n_{control}}\sum_{j=1}^{n_{control}} q_{control,j}, \tag{4.6}$$

$$A = \frac{1}{n}\left(\sum_{i=1}^{n_{treatment}} q_{treatment,i} + \sum_{j=1}^{n_{control}} q_{control,j}\right), \tag{4.7}$$

where $n_{treatment}$ is the number of treatment replicates, $n_{control}$ is the number of control replicates, $n = n_{treatment} + n_{control}$, $q_{treatment,i}$ is the log ratio of treatment replicate i and $q_{control,j}$ is the log ratio of control replicate j. With these definitions we give the multiple-replicates normalizations, defined analogously to those in the log ratio type experiment.

Multiple slide normalizations

We again use loess to allow dependence of the mean normalization of M on A in the following way:

$$M_{norm} = M - c(A) \tag{4.8}$$

where $c(A) = loess(M \sim A)$, with the recommended span for the loess smoother being between 60% and 80%.

For the variance normalization we again use the information about the variance contained in the replicates to get a robust estimator of the overall variance. We calculate the variance of log ratios across the $n_{control}$ replicates in the control dataset and call this $V_{control}$. Similarly we calculate the variance of log ratios across the $n_{treatment}$ replicates in the treatment dataset and call this $V_{treatment}$. Our estimate for the standard deviation s, in M for each gene is given by

$$s = \sqrt{\frac{V_{treatment}}{n_{treatment}} + \frac{V_{control}}{n_{control}}}. \tag{4.9}$$

We then develop the variance normalization similarly to the previous log ratio type experiment case. The variance normalization is given by

$$M_{normv} = \frac{M_{norm}}{s} 1_{[|M_{norm}|<s]} + \frac{M_{norm}}{k} 1_{[|M_{norm}|\geq s]}. \tag{4.10}$$

4.2.3 Summary of model and normalizations for different experiments

A summary of the quantities of interest (used in the normalizations and normal uniform mixture model) and the normalizations is given in Table 4.2.3.

4.2.4 Methods for comparison with NUDGE

We now give brief descriptions of the methods for finding differentially expressed genes that will be used for comparison with NUDGE in the datasets examined in the Results section.

Table 4.1: Summary of Normalization Methods for Different Set-ups

Type of Experiment	Multiple Replicates?	Dye Swap?	Quantity of Interest	Mean Normalization	Variance Normalization
Sample 1 = Red, Sample 2 = Green	No	NA	$\log\left(\frac{Red}{Green}\right)$	Equation 4.3	Equation 4.4
Sample 1 = Red, Sample 2 = Green	Yes	No	$\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{Red_i}{Green_i}\right)$	Equation 4.3	Equation 4.5
Sample 1_{n_1} = Red, Sample 1_{n_2} = Green, Sample 2_{n_1} = Green, Sample 2_{n_2} = Red	Yes	Yes	$\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{Sample1_i}{Sample2_i}\right)$	NA	Equation 4.5
Sample 1 & 2 = Red, Reference = Green	Yes	NA	M, Equation 4.6	Equation 4.8	Equation 4.10

Rule of Two

This simple but popular method, mentioned in [64], involves examining the ratios or average ratios of the two channels for each gene, and calling those genes with a ratio or average ratio greater than two or less than half, differentially expressed. It requires some initial normalization and its performance can depend on the normalization.

t test and adjusted t test

One of the most obvious first approaches to try for this problem is the classical t test, as used, for example, in [3]. A simple normalization consisting of centering the mean of the log ratios within each replicate is often used in this case. One needs to be able to estimate the standard deviations as well.

Because of the large number of tests being run (thousands in the usual cDNA experiment setup), the standard t test needs to be modified to account for the multiple testing. Traditionally the most popular adjustment has been the Bonferroni correction, as mentioned in [50]. For the Bonferroni correction with N genes/tests and significance level α, we instead call each test significant only if it is significant at the $\frac{\alpha}{N}$ level, controlling for the probability of one or more false positives.

EBarrays

This follows a hierarchical Bayes approach for modeling the gene expression levels as detailed in [43]. As in our approach, the data are assumed to be generated by a two-component mixture model, one component for differentially expressed and the other for non-differentially expressed genes, each with their own distribution. The parameters specifying these distributions are estimated from the data, whence the name Empirical Bayes.

Results in this framework are given for two different parametric models in [43]. In the first model, the observed intensities for the replicates in each channel are assumed

to be independently generated from a gamma distribution with a channel-specific scale parameter. The scale parameters are, in turn, assumed to have an inverse gamma distribution, whose parameters are estimated from the whole dataset. In the second model, the log ratios are assumed to be normally distributed, with gene-specific means that are themselves normally distributed. To normalize, the authors divided the log ratio for a given gene and replicate by the average log ratio across genes for that replicate.

Significance Analysis of Microarrays (SAM) [68].

The statistic used to test for differential expression is a regularized t statistic, i.e. the mean value divided by the sum of the standard deviation and a constant. SAM controls the False Discovery Rate (FDR), i.e. the proportion of genes declared to be differentially expressed that are not in truth differentially expressed. A rejection region is fixed and SAM uses a permutation analysis to estimate the FDR. The user then decides on an acceptable rejection region based on their preferences for FDR.

4.3 Real Data Examples

4.3.1 HIV dataset

The HIV dataset that we analyze consists of four replicate experiments comparing cDNA from CD4+ T cell lines at 1 hour after infection with HIV-1BRU with non-infected cell lines on each slide; see [70] for details. There were four slides in total with the same RNA preparations hybridized to each. This dataset is useful in testing the specificity and sensitivity of methods for identifying differentially expressed genes, since there are 13 genes known to be differentially expressed (spots containing PCR products from segments of the HIV-1 genome which the cDNA of the infected cells should hybridize to and the non-infected should not) called positive controls, and 29 genes known not to be (non-human genes which neither infected nor non-infected

cDNA samples should hybridize to) called negative controls.

There are 4608 gene expression levels recorded in each replicate. The four replicates have balanced dye swaps, so no mean normalization of the (averaged across replicates) log ratios was necessary provided we always used one sample (say the infected sample) in the numerator of the log ratio and the other (non-infected sample) in the denominator regardless of which dye was used to label which sample in each array/slide.

NUDGE took a few seconds to run. All 13 positive controls, no negative controls and three other genes were found to be differentially expressed (with posterior probability greater than 0.5).

It is clear from Figure 4.1 that the rule of two under any normalization gave less than optimal results. In all cases the rule of two correctly found the positive control genes to be differentially expressed. However, in the unnormalized case it also incorrectly found 3 of the 29 negative controls to be differentially expressed, as well as 58 other genes (including the three found by NUDGE). In the variance-normalized case, it incorrectly found one of the 29 negative controls to be differentially expressed, as well as 27 other genes (including the three found by NUDGE). Even though the rule of two is suboptimal, its performance can be improved through the use of the normalization methods suggested here.

Table 4.2 shows the results of different methods for the control genes. NUDGE had a perfect result for these genes, with no false positives and no false negatives. The Bonferroni-corrected t test was the only method considered that recorded any false negatives. The rule of two (normalized or unnormalized), SAM and the EBarrays Lognormal-Normal model all had false positives. Only the EBarrays Gamma-Gamma model equaled NUDGE's performance on these control genes.

In order to assess the stability of the different methods, the four replicates were split into two different subsets of two replicates each (still with balanced dye swaps), and the agreements and disagreements between the genes found to be differentially

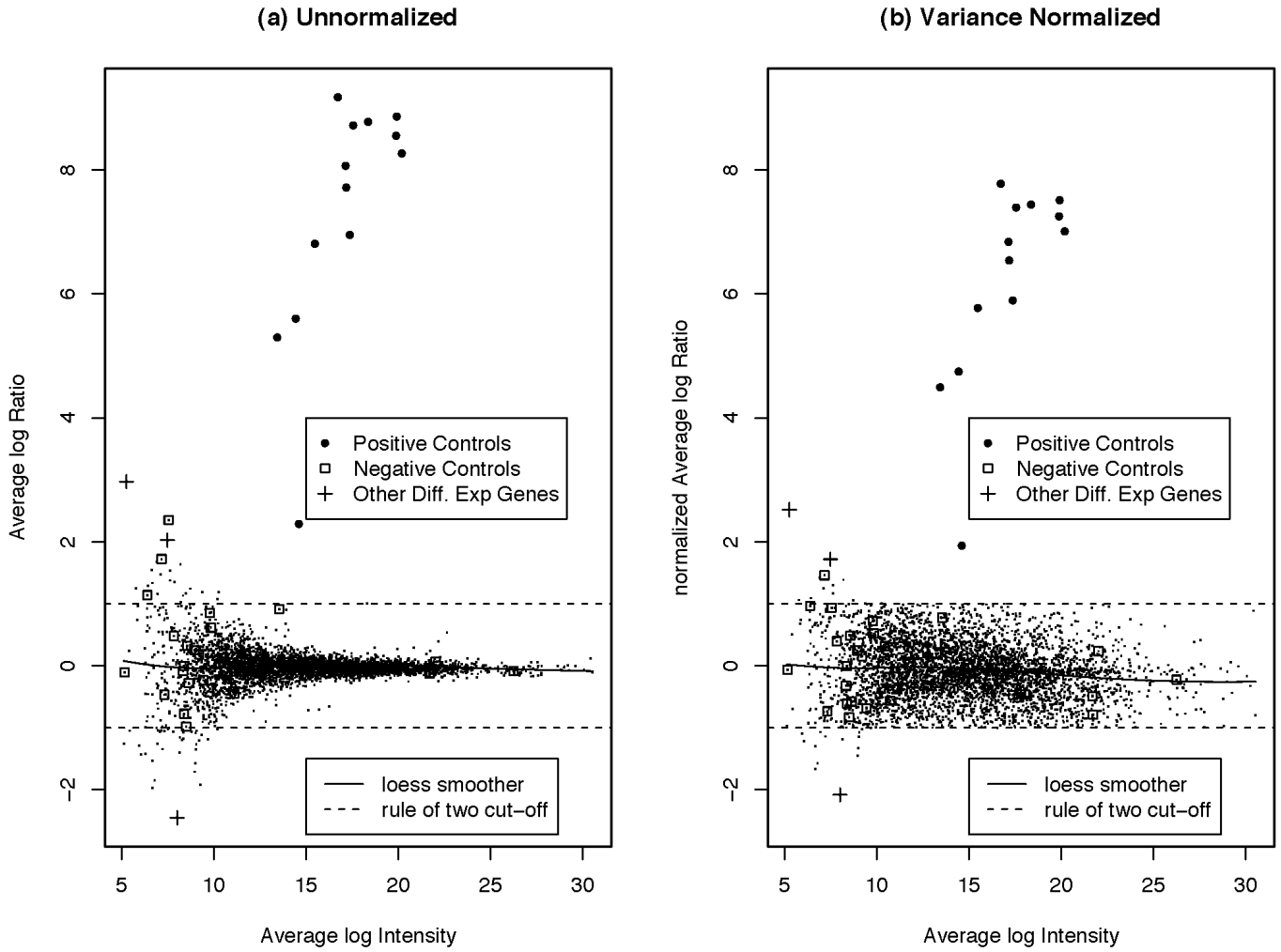

Figure 4.1: Different normalizations of HIV data: (a) raw data, (b) data normalized with respect to the variance. The bullets are the positive controls: NUDGE correctly found them all to be differentially expressed. Other genes found to be differentially expressed by NUDGE are indicated by a plus sign, and all genes found not to be differentially expressed by NUDGE are shown by small dots. Negative controls are indicated by a box. No negative controls were found to be differentially expressed by NUDGE.

Table 4.2: Summary of Results for HIV data for control genes

Method	Number of False Negatives	Number of False Positives
Rule of Two (on unnormalized data)	0	3
Rule of Two (on variance normalized data)	0	1
NUDGE	0	0
SAM	0	2
EBarrays (GG)	0	0
EBarrays (LNN)	0	1
t test	0	1
Bonferroni corrected t test	1	0

expressed in each of the two datasets was calculated for each of the methods. A summary of the results is given in Table 4.3. The number of genes found to be differentially expressed in each of the datasets by each method is given in Table 4.4.

Table 4.3: Number of agreements and disagreements between the differentially expressed genes found in the two sets of two replicates for the HIV data

	NUDGE	SAM	EBarrays GG	EBarrays LNN	t test	Bonferroni t test
Agreements	14	19	13	13	34	15
Disagreements	27	153	16	32	531	217

Table 4.4: Number of genes declared to be differentially expressed by each method for the HIV data using 2 and 4 replicates

	NUDGE	SAM	EBarrays GG	EBarrays LNN	t test	Bonferroni t test
All 4 replicates	16	42	24	19	26	12
Replicates 1&3	30	49	23	27	193	83
Replicates 2&4	25	142	19	31	406	164

Comparison of results depends on how one weights agreement (roughly indicating true positives) against disagreement (roughly indicating false positives). NUDGE had more agreement and less disagreement than EBarrays-LNN, and thus dominated it on both these criteria. The t test, both raw and corrected, and SAM, had more agreement, but at the cost of a much higher level of disagreement than NUDGE. NUDGE had more agreement, but also significantly more disagreement, than EBarrays with a Gamma-Gamma model.

Finally in order to check the empirical fit of the model to this data (where we know we have both differentially and non-differentially expressed genes) we plot the model's fitted density over a histogram of the normalized log ratios in Figure 4.2. The model seems to fit the normalized data fairly well.

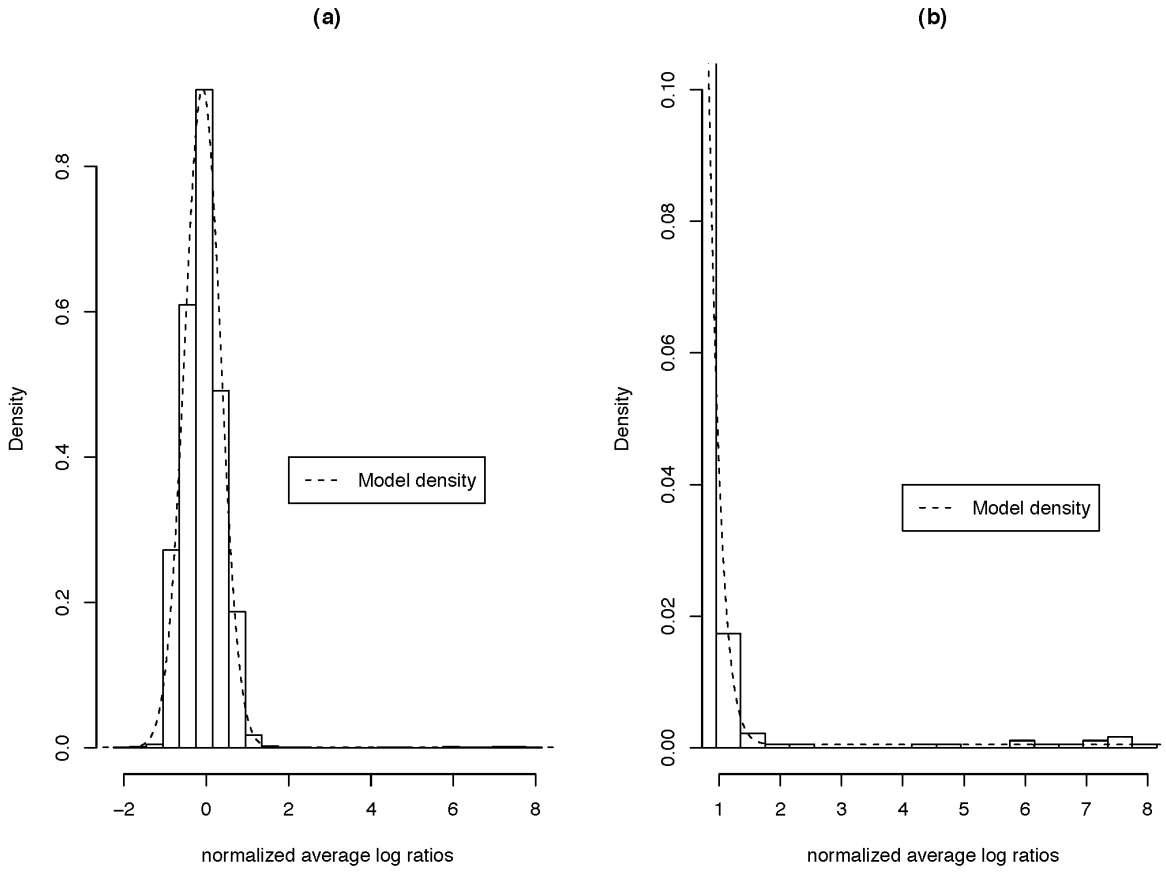

Figure 4.2: Plot (a) shows a histogram of the normalized average log ratios for the HIV data along with a dashed line showing the model-fitted density. Plot (b) shows a close-up of the right-hand tail of the histogram (where the positive controls lie) with a dashed line showing the model-fitted density.

4.3.2 Like-Like dataset

This dataset is from a microarray experiment where the same samples (with different dyes) were hybridized to an array with 7680 genes. The expression levels in the red and green dyes were extracted from the image using customized software written at the University of Washington (Spot-On Image, developed by R. E. Bumgarner and Erick Hammersmark). The genes should be equally highly expressed, as each sample is the same, so ideally we should find few differentially expressed genes.

Figure 4.3 (a) shows the log ratios plotted against the log total intensities. Here we see evidence of the dye effect, since if it were not present the data would fall with some variation about a zero-intercept horizontal line. Figure 4.3 (b) is a plot of the mean-normalized log ratio against the log total intensity. In Figure 4.4 we plot the absolute mean-normalized log ratio as a function of log total intensity. We use a loess smoother of this as a robust estimate of how spread depends on log total intensity. This is used to get the loess variance-normalized log ratios, which are plotted against the log total intensities in Figure 4.3 (c). The data now look much more normal and homoscedastic. The NUDGE method took less than 5 seconds to run with 10 iterations of the EM algorithm.

The results are summarized in Table 4.5. NUDGE found 28 differentially expressed genes (with posterior probability greater than 0.5). This is a false positive rate of 0.4%. With no normalization, the rule of two declared 3233 genes to be differentially expressed, 42.1% of the total; clearly this is not appropriate. After the data had been mean-normalized, the rule of two found 281 differentially expressed genes, a false positive rate of 3.7%. When the data have been mean- and variance-normalized, the rule of two finds 105 genes, a false positive rate of 1.4%, still higher than NUDGE. Since there is only one replicate in this case, neither t tests, SAM nor EBarrays can be used to test for differential expression.

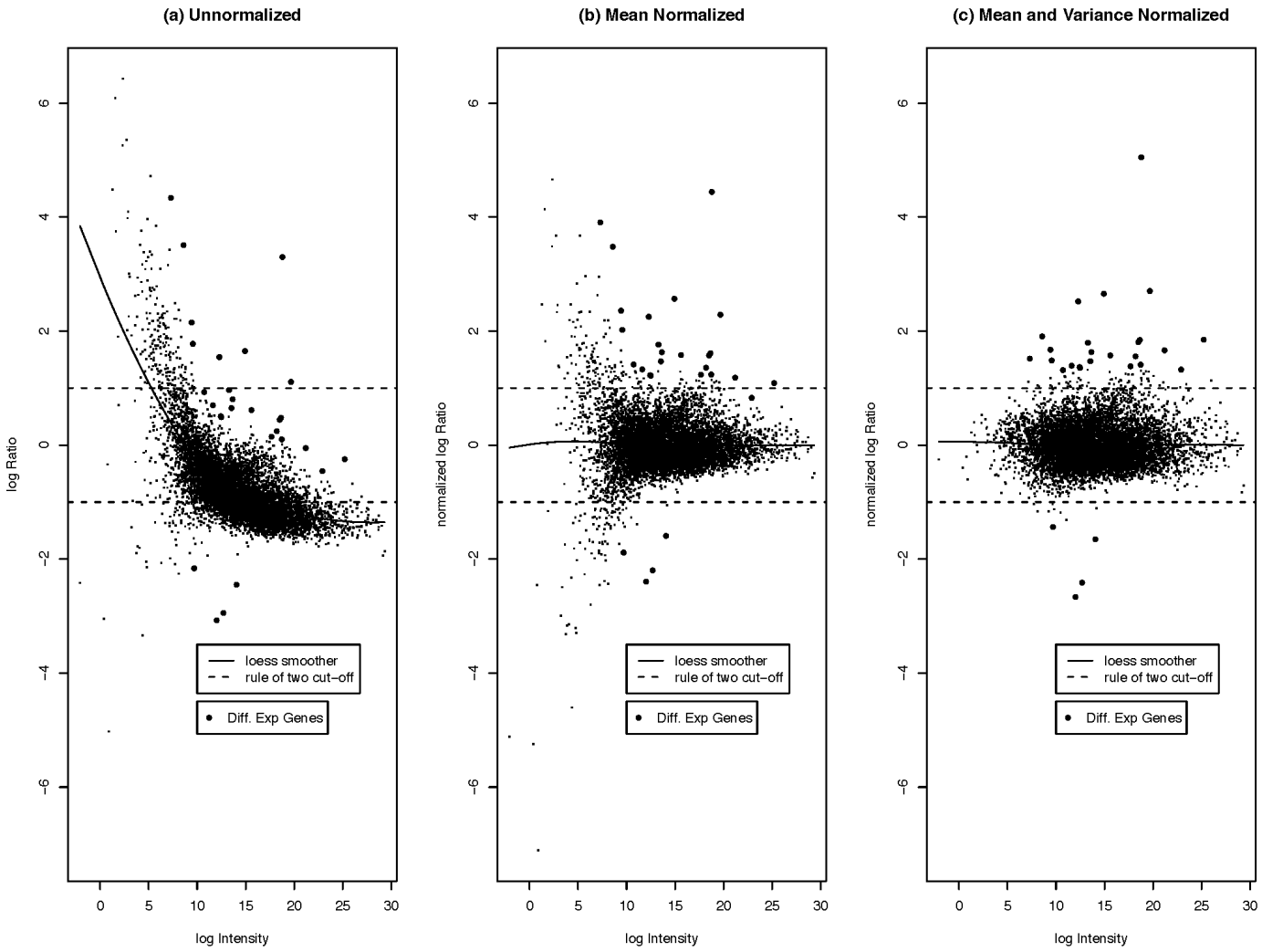

Figure 4.3: Different normalizations of Like-like data: (a) raw data, (b) data normalized with respect to the mean, (c) data normalized with respect to both mean and variance. Diff. Exp. Genes are genes found to be differentially expressed by NUDGE (with posterior probability greater than 0.5).

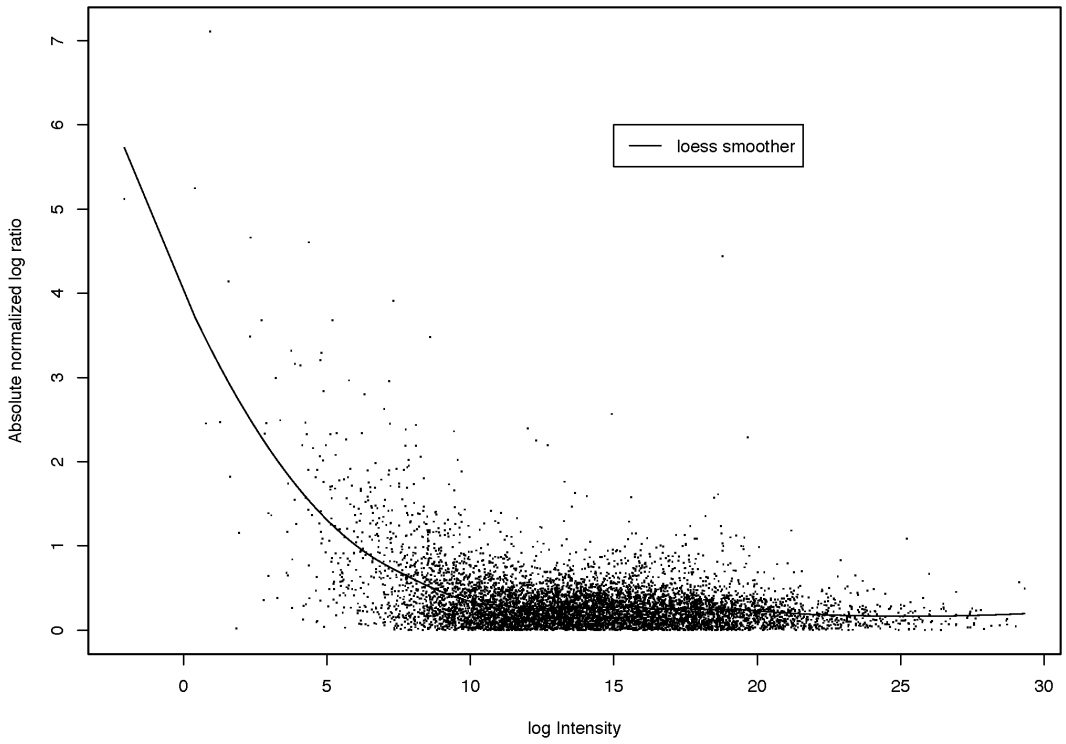

Figure 4.4: Absolute mean normalized log ratio versus log total intensity for Like-like Data. The loess line in this plot represents the estimate of the gene-specific Mean Absolute Deviation (MAD), a robust estimator of spread.

Table 4.5: Results for the Like-like data - SAM, EBarrays and t tests are not applicable to single slide data.

Method	Estimated False Positive Rate
Rule of Two (on unnormalized data)	42.1%
Rule of Two (on mean loess normalised data)	3.7%
Rule of Two (on mean and variance loess normalised data)	1.4%
NUDGE	0.4%

4.3.3 Apo AI dataset

This dataset was analyzed in [26] and 8 genes were suggested to be differentially expressed. The data was obtained from 8 mice with the Apo AI gene knocked out and 8 normal mice. However the replicates were not created simply by comparing samples from control labeled with one dye versus knock-out mice labeled with the other. Instead, cDNA was created from samples from each of the 16 mice (both control and knock-out) and labeled with a red dye. The green dye was used in all cases on cDNA created by pooling all 8 control mice. The statistic used in [26] was

$$\frac{\text{average of knock-out log ratios - average of control log ratios}}{\sqrt{\frac{1}{8}(\text{Variation of knock-out log ratios + Variation of control log ratios})}}. \tag{4.11}$$

We used the numerator of this statistic, which is the same as M defined in equation 4.6, in place of ordinary average log ratios, as detailed in the Methods section. Again the method took only a few seconds to run. Figure 4.5 shows the data at different stages of normalization along with the genes found to be differentially expressed in [26]. Table 4.6 shows the gene position numbers of those genes whose posterior probability of being differentially expressed was in the top sixteen found by NUDGE. All eight of the genes found by [26] to be differentially expressed were also found to be differentially expressed with high probability by our method. The lines in Figure 4.5 indicating the rule of two cut-off appear either to miss genes that are differentially expressed (in the unnormalized and mean-normalized cases), or to give a large number of possible false positives (in the mean- and variance-normalized case).

For application of SAM, the data were normalized in the standard way, by centering the log ratios across genes within a replicate about zero. Two different levels of the SAM control parameter delta gave reasonable answers when using SAM on this data set. The first level (0.61) found 15 genes to be differentially expressed, including the eight genes found in [26] and by NUDGE, and the False Discovery Rate was estimated to be 5.3%. If we assume that only these eight genes are correct, this would actually correspond to a False Positive Rate of 46.7%. The second level (3.53) found

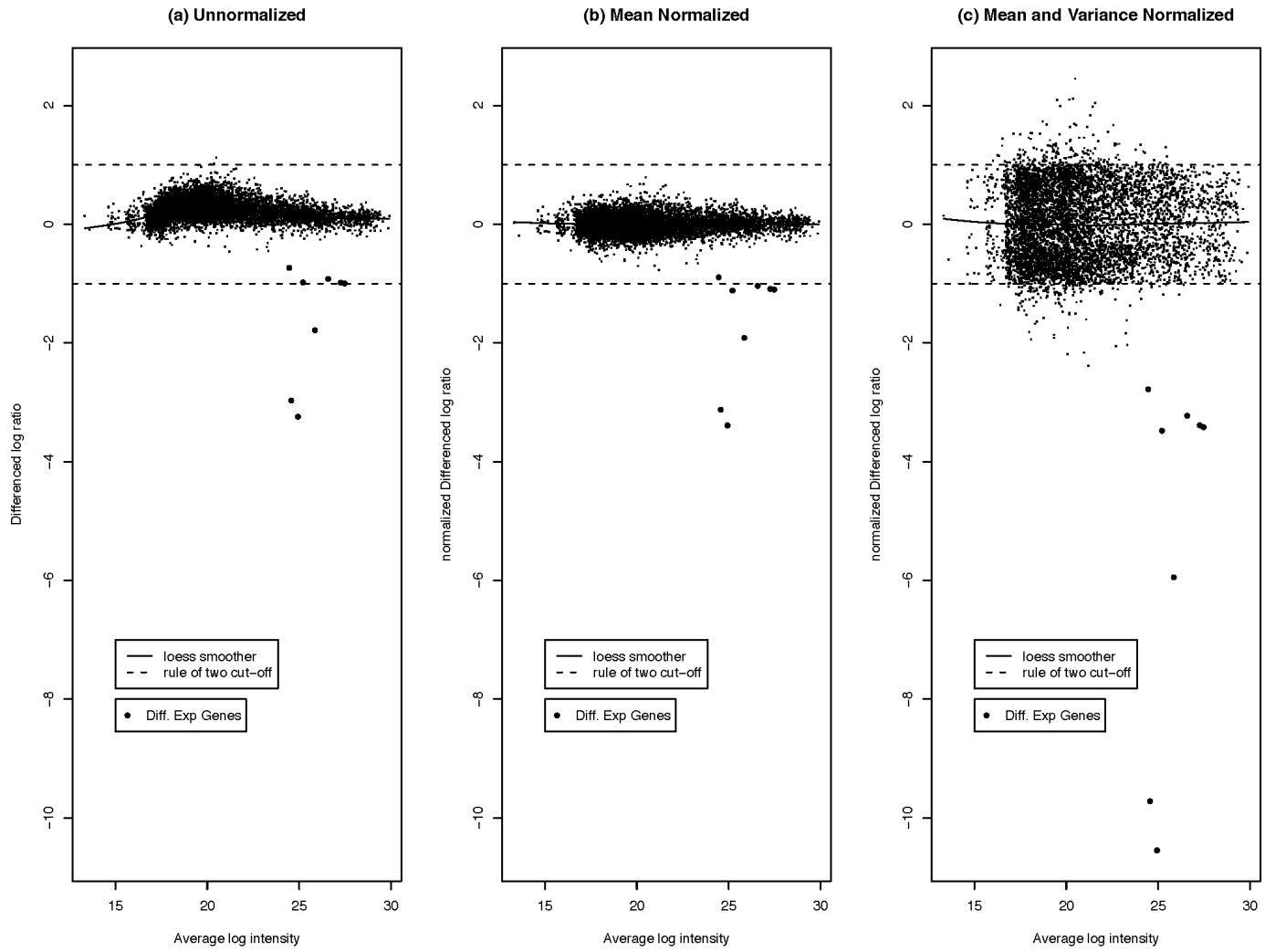

Figure 4.5: Different Normalizations of Apo data: (a) raw data , (b) data normalized with respect to the mean, (c) data normalized with respect to both mean and variance. Diff. Exp. Genes are genes found to be differentially expressed by NUDGE (with posterior probability greater than 0.5).

Table 4.6: NUDGE's Top 16 Genes from the Apo data

Top 16 genes in terms of NUDGE posterior probability of differential expression		
Row numbers in data matrix	Probability of differential expression	Found by Dudoit et al [26]?
540	1.000	Yes
2149	1.000	Yes
5356	1.000	Yes
1739	0.999	Yes
4139	0.999	Yes
2537	0.998	Yes
4941	0.993	Yes
1496	0.829	Yes
5986	0.330	No
541	0.263	No
716	0.099	No
2538	0.087	No
1224	0.066	No
799	0.060	No
1204	0.057	No
3729	0.050	No

six genes to be differentially expressed, a subset of the eight genes found by [26], and the False Discovery Rate was estimated to be 13.3%. Assuming that only those eight genes are correct, this corresponds to a False Positive Rate of 0% but a False Negative Rate of 25%. These were the best results we obtained using SAM.

For similarly normalized data, both the t test and the Bonferroni adjusted t test found the 8 genes identified by [26] to be differentially expressed. However, the t test found an additional 852 genes to be differentially expressed at the 5% significance level (13.5% of all genes), and the Bonferroni adjusted t test found an additional two genes to be differentially expressed. A summary of the results for the Apo data is given in Table 4.7.

4.4 Conclusions

We have proposed a simple method for detecting differentially expressed genes that is fast and can be applied to single-slide and multiple-replicate experiments, as well as to log ratio difference experiments. It accounts for the multiple comparisons involved, and produces a posterior probability of differential expression for each gene, rather than just a yes/no testing result. The posterior probabilities can be used either to declare which genes are differentially expressed, or to produce a ranked list of genes for further analysis. The method worked well for the three datasets that we analyzed. In terms of known false positives and false negatives, the method outperforms all multiple-replicate methods except for the Gamma-Gamma EBarrays method to which it offers comparable results with the added advantages of greater simplicity, speed, fewer assumptions and applicability to the single replicate case.

Our method can be seen as a parametric alternative to adjustment of tests for multiple comparisons using false discovery rate ideas [65], or empirical Bayes formulations [28]. A similar idea was proposed in [19] for large numbers of tests, in which the distribution of the test statistic was modeled as a mixture of two normals, one corresponding to the null hypothesis being true, and the other to its being false. This

Table 4.7: Results for the Apo data

Method	Number of 8 Dudoit et al [26] genes found to be differentially expressed	Number of other genes found to be differentially expressed
Rule of Two (on unnormalized data)	3	0
Rule of Two (on mean normalized data)	7	0
Rule of Two (on mean and variance normalized data)	8	134
NUDGE	8	0
SAM (delta=0.61)	8	7
SAM (delta=3.53)	6	0
t test	8	852
Bonferroni corrected t test	8	2

differs from our approach in that we use a uniform distribution for the mixture component that corresponds to departures from the null, rather than a mean-shifted normal. Because of this, a method such as [19] could not find both over- and underexpressed genes.

A similar idea with different distributional assumptions, using only normally distributed components is given in [59]. Instead of the average log ratios used in the method presented in this chapter, [59] use a t-type statistic using the difference of average gene intensities. A more complex approach given by [8] involves modeling each level of differential expression with its own normal component.

In our approach the important aspect of the mixture is the cutoff points where the weighted normal density falls below the height of the weighted uniform density. Points beyond the cutoff are declared to be differentially expressed (under a 0.5 posterior probability rule). These cut-off points are relatively unaffected by outliers which affect the range of the data and thus the range and height of the uniform component, because the normal density falls off very rapidly towards the tails, and also because the estimated mixture weights change accordingly.

An important part of the method is normalization in terms of variance as well as mean. This extends the original lowess normalization in [26]. As a preprocessing step, it improves the performance not only of NUDGE, but also of other methods, including the simplest of all, the rule of two. Thus, this normalization method may be useful as a preprocessing tool for analysis of differential gene expression, regardless of which method is used to draw final inferences.

Appendix: Different Distributions for non-differentially expressed genes

One of the more useful aspects of the like-like data is that given we know there *should* be no differentially expressed genes present, i.e. all the data, once normalized, should be distributed normally; we can check the validity of this assumption. If we look at theoretical versus sample quantile plots to examine the normality of the like-like data we see from Figure 4.6 that while the normalizations improve the normality of the data some non-normal aspects remain.

The most obvious distribution to examine after the normal is the Student's t distribution for varying numbers of degrees of freedom. We look at these in Figures 4.7, 4.8, 4.9, 4.10 and 4.11. It would appear from these plots that a t distribution with 6 to 10 degress of freedom does a better job of modeling the normalized log ratios than the normal distribution.

We maximize the log-likelihood of the normalized log ratios for the like-like data with respect to the location, scale and degrees of freedom parameters of the generalized t distribution. The maximum is achieved at location -0.02 (which makes sense given the normalization), scale 0.31 and degrees of freedom 7.81. The maximized log likelihood is -2983 (which is 21 points higher than the maximised log likelihood for the normal-uniform mixture).

The data is then modeled using a (generalized) t-uniform mixture and the maximum likelihood parameters are found. The estimated parameters, as well as those for the normal-uniform mixture are given in Table 4.8. The larger degrees of freedom found for the t makes sense, as the uniform is taking account of some of the extreme observations. The number of false positives (genes incorrectly identified as differentially expressed) for the t-uniform mixture is 7 which is a reduction of 22 from the normal-uniform model. The maximum log-likelihood is also higher for the t-uniform than for the normal-uniform.

Given that we know that this model reduces the number of false positives, we

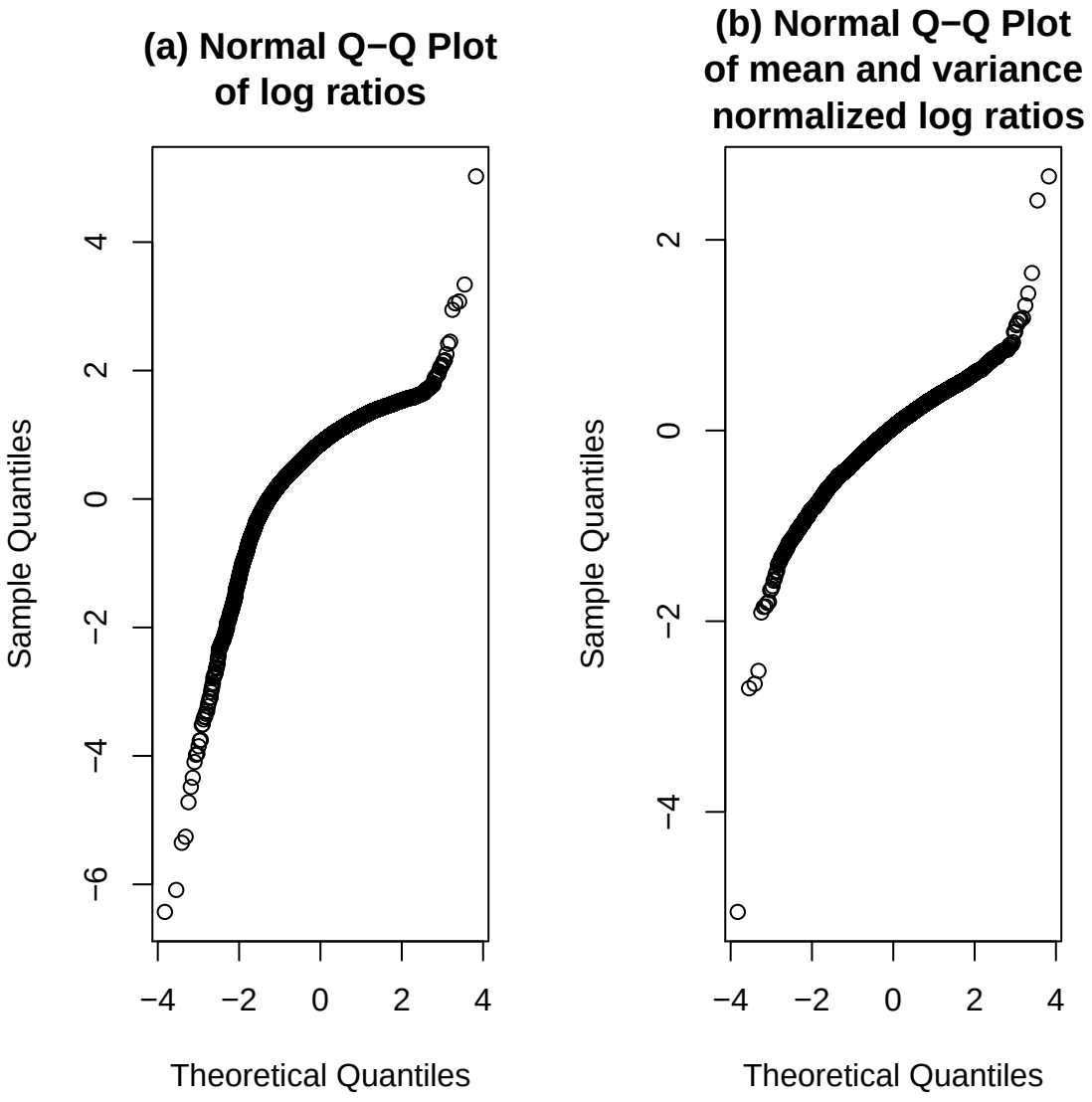

Figure 4.6: (a) Normal Quantile-Quantile plot for non-normalized log ratios, (b) Normal Quantile-Quantile plot for mean and variance normalized log ratios

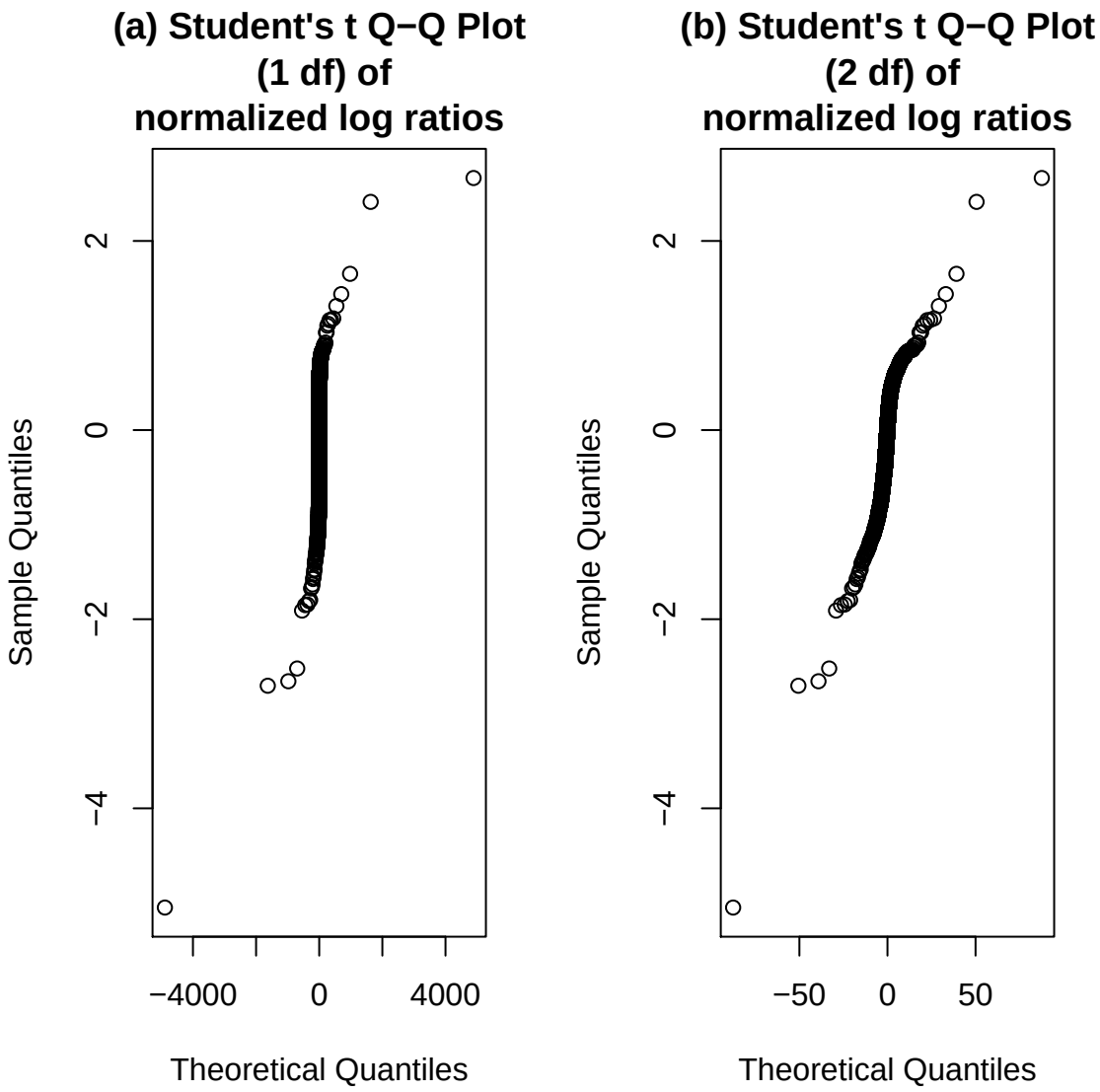

Figure 4.7: t-distributed Quantile-Quantile plots (1 & 2 degrees of freedom) for normalized like-like data. (a) 1 degree of freedom, (b) 2 degrees of freedom

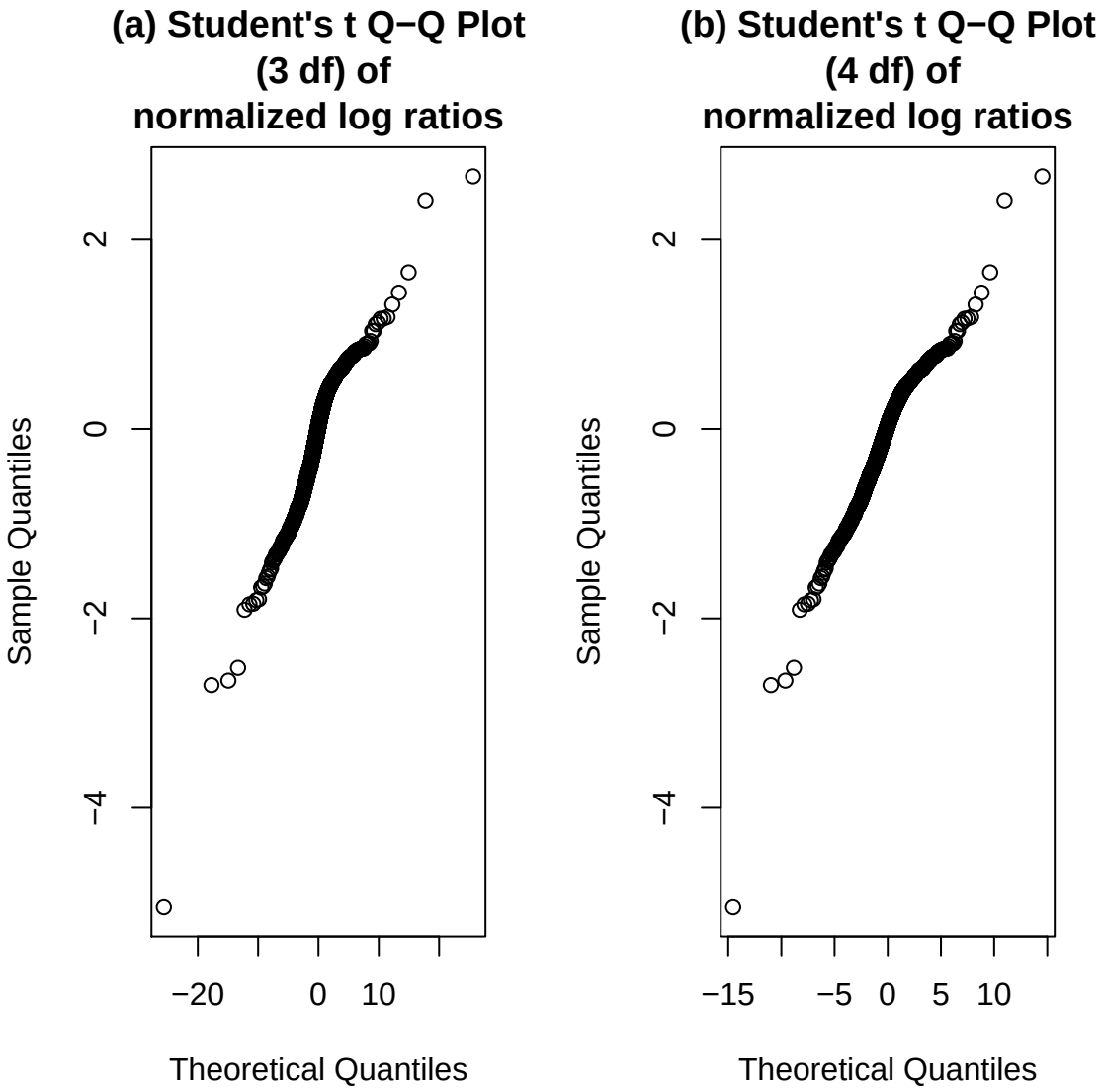

Figure 4.8: t-distributed Quantile-Quantile plots (3 & 4 degrees of freedom) for normalized like-like data. (a) 3 degrees of freedom, (b) 4 degrees of freedom

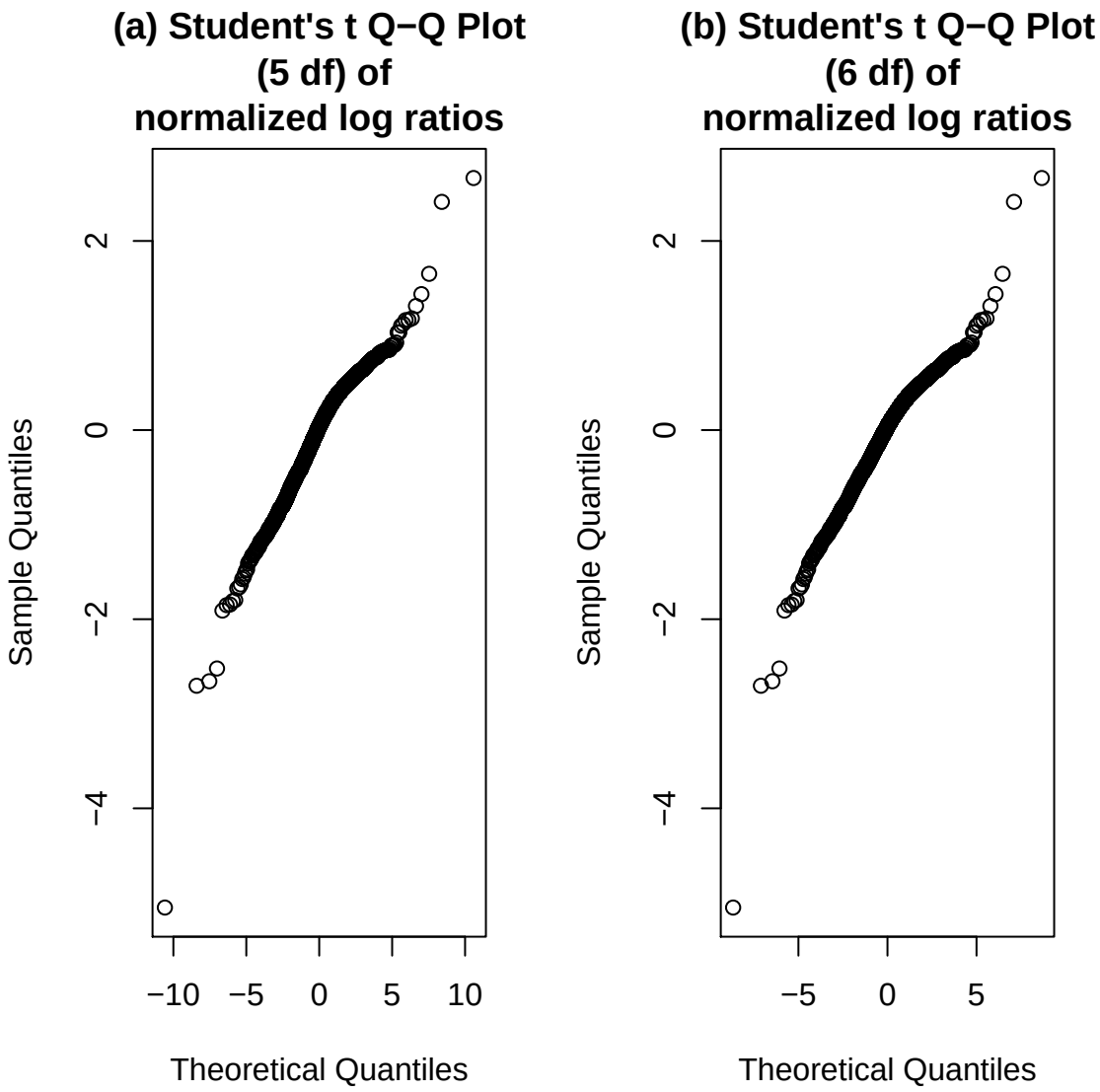

Figure 4.9: t-distributed Quantile-Quantile plots (5 & 6 degrees of freedom) for normalized like-like data. (a) 5 degrees of freedom, (b) 6 degrees of freedom

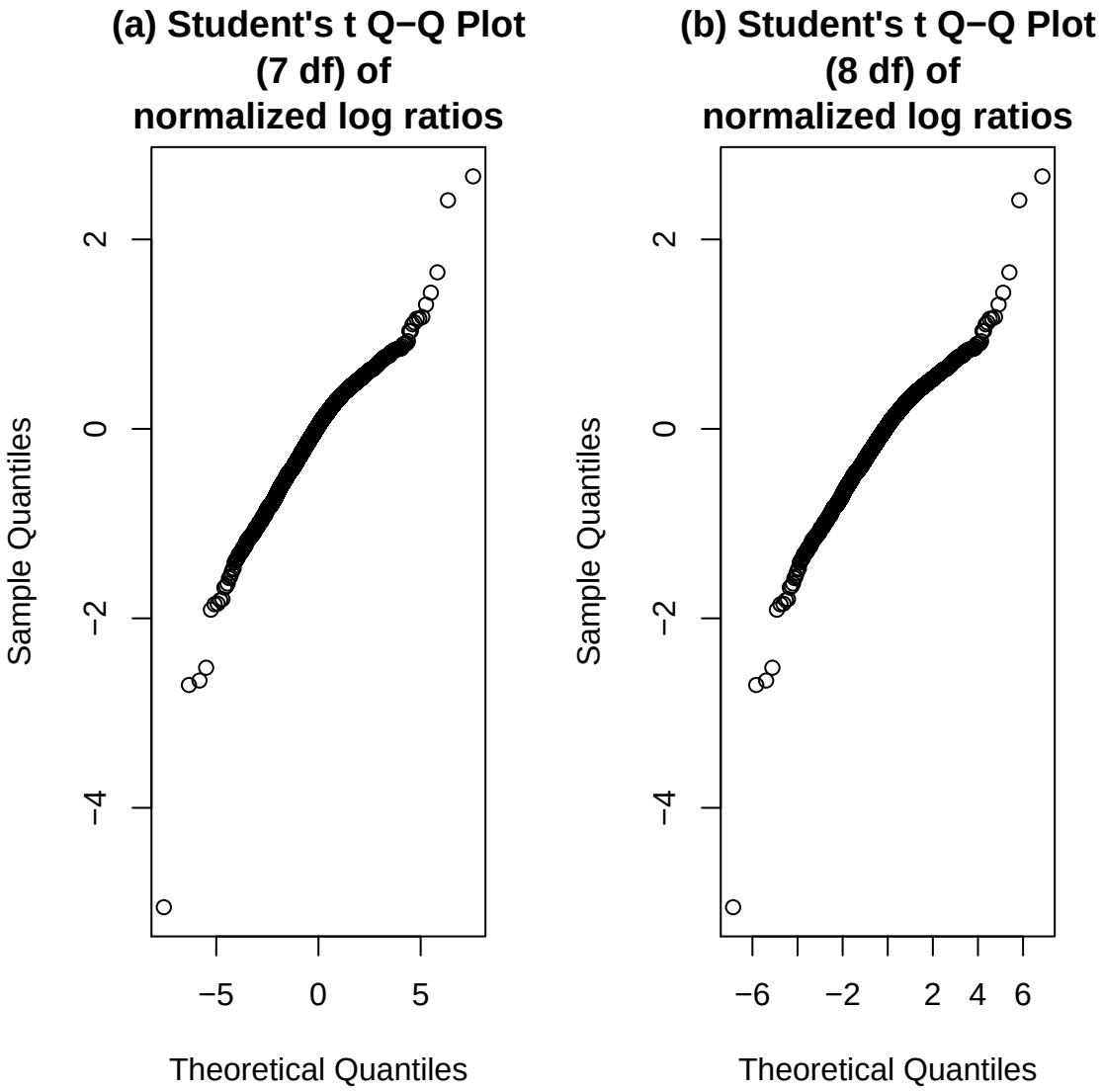

Figure 4.10: t-distributed Quantile-Quantile plots (7 & 8 degrees of freedom) for normalized like-like data. (a) 7 degrees of freedom, (b) 8 degrees of freedom

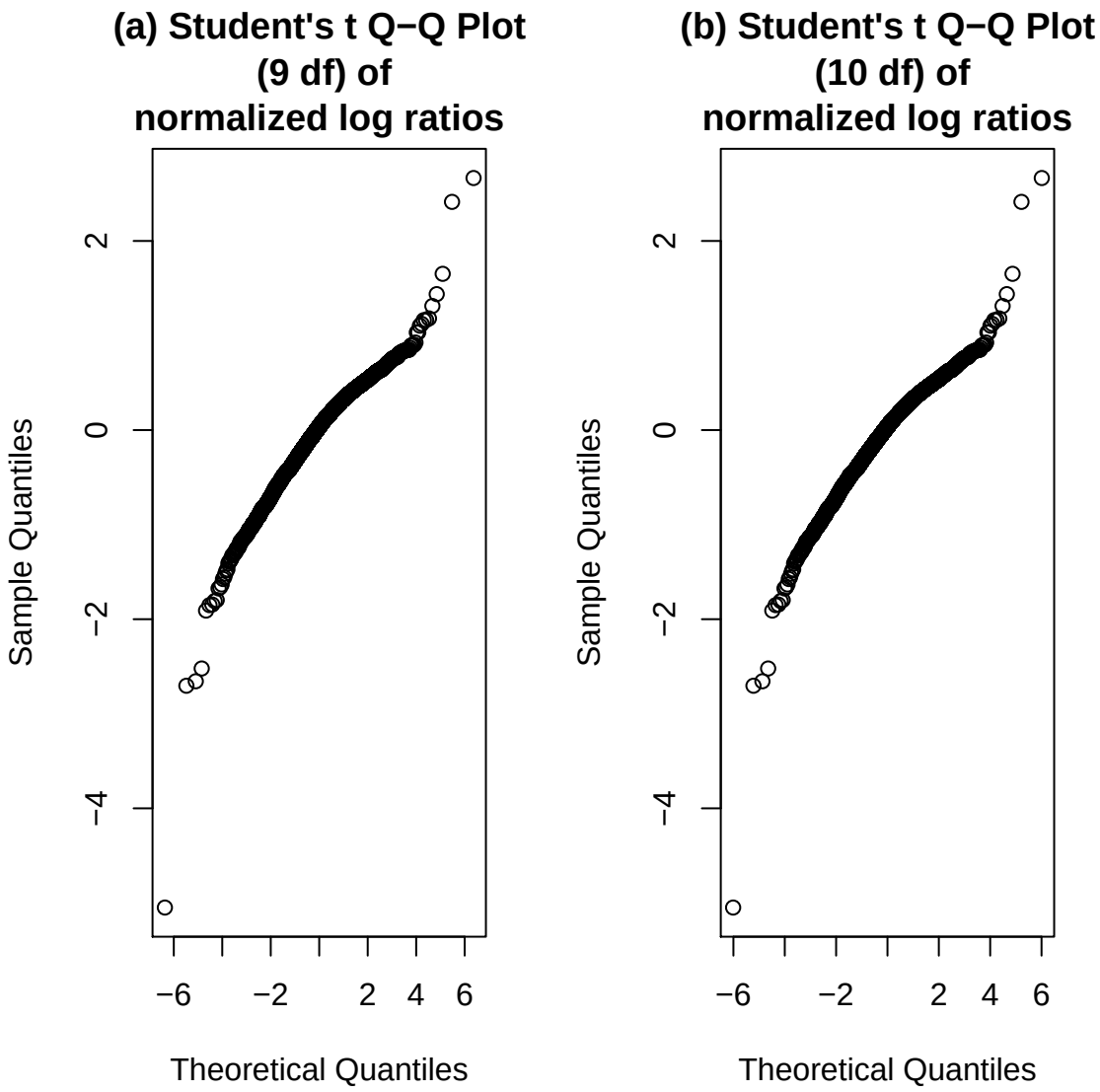

Figure 4.11: t-distributed Quantile-Quantile plots (9 & 10 degrees of freedom) for normalized like-like data. (a) 9 degrees of freedom, (b) 10 degrees of freedom

Table 4.8: Results for Maximum Likelihood Estimation of the Mixture Models for the like-like data

Model	Location	Scale	Degrees of Freedom	Mixing Proportion	Max. Log Likelihood	# of False Positives
Student's-t-Uniform	-0.02	0.32	10.61	0.9978	-2970	7
Normal-Uniform	-0.00	0.35	-	0.9937	-3004	29

need to ensure that it reduces (or does not increase) the number of false negatives. To check this we look at the HIV dataset. We first fit the model allowing the degrees of freedom to be unconstrained and then constrain the degrees of freedom to be the same of those found fitting a single t distribution to the like-like data (7.81). The estimated parameters are given in Table 4.9.

We can see that, unlike the like-like data, when the degrees of freedom for the t in the t-uniform mixture is allowed to be unconstrained it is estimated to be an extremely large number with the resulting model being almost identical to the normal-uniform model. When we constrain the degrees of freedom to be the same as that estimated for the single t distribution fit to the like-like data, the scale parameter becomes somewhat smaller as does the log likelihood and there is an additional false negative that was not present in the other two models.

Table 4.9: Results for Maximum Likelihood Estimation of the Mixture Models for the HIV data. (c) indicates that the parameter is constrained to be a certain value.

Model	Location	Scale	Degrees of Freedom	Mixing Proportion	Max. Log Likelihood	# of (known) False Positives	# of (known) False Negatives	# of genes found to be Differentially Expressed
Student's-t-Uniform	-0.09	0.44	2884582	0.9952	-2878	0	0	16
Student's-t-Uniform (c)	-0.09	0.40	7.81 (c)	0.9961	-2921	0	1	13
Normal-Uniform	-0.09	0.44	-	0.9952	-2878	0	0	16

Chapter 5

FUTURE WORK

5.1 Variable Selection

An obvious extension of overall variable selection for clustering (looking for variables which separate any/all of the clusters in some way) is to look at the idea of each cluster having a set of variables that separates it from the other clusters. This has been looked at in the context of heuristic clustering ([47]) and has been referred to as biclustering, following the idea that the variables are clustered into different (possibly overlapping) sets for separating clusters of observations and the observations are clustered (on the variables). We would like to look at idea of biclustering from a model-based point of view.

Clearly, different models than the ones proposed in chapters 2 and 3 can be used to test for variables' clustering capabilities. The one variable case of $Y^{(?)}$ is simpler and requires less assumptions but we could also have $Y^{(?)}$ being more than one variable. Clearly this will be easy in the latent class set-up since we are modeling with independence between $Y^{(?)}$ and $Y^{(clust)}$ for M_1. However, more assumptions will be needed in the context of model-based clustering. One possibility is to allow each variable in $Y^{(?)}$ to be regressed on both the variables in $Y^{(clust)}$ *and* the other variables in $Y^{(?)}$. This will allow greater flexibility in the kind of search algorithms which can be used.

5.2 Differential Gene Expression Detection

Although we have presented a model that could be applicable to any type of microarray chip technology (Aguilent/Affymetrix/cDNA chips) the normalizations are specific to cDNA arrays. Currently Affymetrix chips are falling in price and becoming

increasingly popular. To make this method applicable to Affymetrix chips we need to adjust normalization procedures for Affymetrix technology.

Alternatively we could remove the normalization stage (except for normalizing in the sense of ensuring comparability of different slides) and use different modeling assumptions to model the non-normalized log ratios directly e.g. using a mixture of normals to model the non-differentially expressed genes group and a uniform or some other type of noise component to model differentially expressed genes. Or we could model both the log ratios and the log total intensities using a mixture with the principal curves technology ([39]) for the non-differentially expressed genes and the uniform or other distributions to model the differentially expressed genes.

BIBLIOGRAPHY

[1] E. Anderson. The irises of the Gaspé Peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

[2] E. Anderson. The specicies problem in *iris*. *Annals of the Missouri Botanical Garden*, 23:457–509, 1936.

[3] Stuart M. Arfin, Anthony D. Long, Elaine T. Ito, Lorenzo Tolleri, Michelle M. Riehle, Eriks S. Paegle, and G. Wesley Hatfield. Global gene expression profiling in Escherichia coli K12. *J. Biol. Chem.*, 275(38):29672–29684, 2000.

[4] Jens Henrik Badsberg. Model search in contingency tables by CoCo. In Y. Dodge and J. Whittaker, editors, *Computational Statistics*, volume 1, pages 251–256, 1992.

[5] Jeffrey. D. Banfield and Adrian E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 48:803–821, 1993.

[6] M. P. Becker and I. Yang. Latent class marginal models for cross-classifications of counts. *Sociological Methodology*, 28:293–326, 1998.

[7] P. Brodatz. *Textures: A Photgraphic Album for Artists and Designers*. Dover Publications Inc., New York, 1966.

[8] Philippe Broët, Sylvia Richardson, and François Radvanyi. Bayesian hierarchical model for identifying changes in gene expression from microarray experiments. *Journal of Computational Biology*, 9(4):671–683, 2002.

[9] Michael J. Brusco and J. Dennis Cradit. A variable selection heuristic for k-means clustering. *Psychometrika*, 66:249–270, 2001.

[10] N. A. Campbell and R. J. Mahon. A multivariate study of variation in two species of rock crab of genus *Leptograpsus*. *Australian Journal of Zoology*, 22:417–425, 1974.

[11] Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

[12] Kaushik Chakrabarti and Sharad Mehrotra. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *The VLDB Journal*, pages 89–100, 2000.

[13] W. C. Chang. On using principal components before separating a mixture of two multivariate normal distributions. *Applied Statistics*, 32:267–275, 1983.

[14] Y Chen, E. R. Dougherty, and M. L. Bittner. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *J. Biomed. Optics*, 2:364–374, 1997.

[15] D. M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 29:181–212, 1997.

[16] W. S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, 74:829–836, 1979.

[17] W. S. Cleveland. Locally-weighted regression: An approach to regression analysis by local fitting. *J. Am. Stat. Assoc.*, 83:596–610, 1988.

[18] C. C. Clogg and L. A. Goodman. Latent structure analysis of a set of multidimensional contingency tables. *Journal of the American Statistical Association*, 79:762–771, 1984.

[19] D.R. Cox and Man Yu Wong. A simple procedure for the selection of significant effects. *J. R. Stat. Soc. Ser. B*, 66:395–400, 2004.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Stat. Soc. Ser. B*, 39(1):1–38, 1977.

[21] Wayne S. Desarbo, J. Douglas Carroll, Linda A. Clarck, and Paul E. Green. Synthesized clustering: A method for amalgamating clustering bases with differential weighting of variables. *Psychometrika*, 49:57–78, 1984.

[22] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64:304–310, 1989.

[23] Mark Devaney and Ashwin Ram. Efficient feature selection in conceptual clustering. In *Machine Learning: Proceedings of the Fourteenth International Conference*, pages 92–97, Nashville, TN, 1997.

[24] Chris Ding, Xiaofeng He, Hongyuan Zha, and Horst D. Simon. Adaptive dimension reduction for clustering high dimensional data. In *Proceedings of IEEE International Conference on Data Mining*, pages 147–154, Maebashi, Japan, 2002.

[25] Sandrine Dudoit, Juliet Shaffer, and Jennifer Boldrick. Multiple hypothesis testing in microarray experiments. *Stat. Sci.*, 18(1):71–103, 2003.

[26] Sandrine Dudoit, Yee Hwa Yang, Matthew Callow, and Terence Speed. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sin.*, 12:111–139, 2002.

[27] Jennifer G. Dy and Carla E. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proceedings of seventeenth International Conference on Machine Learning*, pages 247–254. Morgan Kaufmann, San Francisco, CA, 2000.

[28] B. Efron, R. Tibshirani, J.D. Storey, and V. Tusher. Empirical Bayes analysis of a microarray experiment. *J. Am. Stat. Assoc.*, 96:1151–1160, 2001.

[29] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[30] C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611–631, 2002.

[31] Chris Fraley and Adrian E. Raftery. Enhanced software for model-based clustering. *Journal of Classification*, 20:263–286, 2003.

[32] Jerome H. Friedman and Jacqueline J. Meulman. Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society, Series B*, 66:to appear, 2004.

[33] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40:11–61, 1989.

[34] G Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. In *Proceedings of the National Academy of Sciences USA, 94*, volume 94, pages 12079–12084, 2000.

[35] R. Gnanadesikan, J. R. Kettenring, and S. L. Tsao. Weighting and selection of variables for cluster analysis. *Journal of Classification*, 12:113–136, 1995.

[36] Leo A. Goodman. Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61:215–231, 1974.

[37] P. E. Green and A. M. Krieger. Alternative approaches to cluster-based market segmentation. *Journal of the Market Research Society*, 37:221–239, 1995.

[38] A. Guérin-Dugué and C. Avilez-Cruz. High order statistics from natural textured images. In *ATHOS Workshop on System Identification and High Order Statistics*, Sophia-Antipolis, France, September 1993.

[39] Trevor Hastie and Werner Stuetzle. Principle curves. *Journal of the American Statistical Association*, 84:502—-516, 1988.

[40] Hosmer and Lemeshow. *Applied Logistic Regression*. Wiley, 1989.

[41] R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[42] Robert E. Kass and Adrian Raftery. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[43] C. M. Kendziorski, M. A. Newton, H. Lan, and M. N. Gould. On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Stat Med*, 22:3899–3914, 2003.

[44] Christine Keribin. Consistent estimate of the order of mixture models. *Comptes Rendues de l'Academie des Sciences, Série I-Mathématiques*, 326:243–248, 1998.

[45] Martin H. Law, Anil K. Jain, and Mário A. T. Figueiredo. Feature selection in mixture-based clustering. In *Proceedings of Conference of Neural Information Processing Systems*, Vancouver, 2002.

[46] Paul F. Lazarsfeld. *Measurement and Prediction*, chapter The Logical and Mathematical Foundations of Latent Structure Analysis, pages 362–412. Princeton University Press, 1950.

[47] Laura Lazzeroni and Art Owen. Plaid models for gene expression data. *Statistica Sinica*, 12:61–86, 2002.

[48] S. Lemeshow, D. Teres, J. S. Avrunin, and H. Pastides. Predicting the outcome of intensive care unit patients. *Journal of the American Statistical Association*, 83:348–356, 1988.

[49] Jun S. Liu, Junni L. Zhang, Michael J. Palumbo, and Charles E. Lawrence. Bayesian clustering with variable and transformation selections. In Jose M. Bernardo, M. J. Bayarri, A. Philip Dawid, James O. Berger, D. Heckerman, A. F. M. Smith, and Mike West, editors, *Bayesian Statistics*, volume 7, pages 249–275. Oxford University Press, 2003.

[50] Anthony D. Long, Harry J. Mangalam, Bob Y. P. Chan, Lorenzo Tolleri, G. Wesley Hatfield, and Pierre Baldi. Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework. *J. Biol. Chem.*, 276(23):19937–19944, 2001.

[51] Andrew McCallum, Kamal Nigam, and Lyle Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169–178, 2000.

[52] G. J. McLachlan and D. Peel. Robust cluster analysis via mixtures of multivariate t-distributions. In P. Pudil A. Amin, D. Dori and H. Freeman, editors, *Lecture Notes in Computer Science*, volume 1451, pages 658–666. Springer-Verlag, Berlin, 1998.

[53] G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

[54] Geoffrey J. McLachlan, R. Bean, and D. Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18:413–422, 2002.

[55] Alan J. Miller. *Subset Selection in Regression*. Number 40 in Monographs on Statistics and Applied Probability. Chapman and Hall, 1990.

[56] Pabitra Mitra, C. A. Murthy, and Sankar K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:301–312, 2002.

[57] M. A. Newton, C. M. Kendziorski, C. S. Richmond, F. R. Blattner, and K. W. Tsui. On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *J. Comp. Biol.*, 8:37–52, 2001.

[58] M. A. Newton and Christina M. Kendziorski. *The Analysis of Gene Expression Data: Methods and Software*, chapter Parametric Empirical Bayes Methods for Microarrays, pages 254–271. Springer, N.Y., 2003.

[59] Wei Pan, Jizhen Lin, and Chap T. Le. A mixture model approach to detecting differentially expressed genes with microarray data. *Funct. Integr. Genomics*, 3(3):117–124, 2003.

[60] Karl Pearson. Contributios to the theory of mathematical evolution. *Philosophical Transactions of the Royal Society of London*, 185:343–414, 1894.

[61] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[62] Dmitry Rusakov and Dan Geiger. Asymptotic model selection for naive bayesian networks. *Journal of Machine Learning Research*, 6:1–35, 2005.

[63] M. Schena, D Shalon, R. W. Davis, and P. Brown. Quantitative monitoring of gene-expression patterns with a complementary-DNA microarray. *Science*, 270:467–470, 1995.

[64] Mark Schena, Dari Shalon, Renu Heller, Andrew Chai, Patrick O. Brown, and Ronald W. Davis. Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci.*, 93:10614–10619, 1996.

[65] J. D. Storey. A direct approach to false discovery rates. *J. R. Stat. Soc. Ser. B*, 64:479–498, 2002.

[66] Luis Talavera. Dependency-based feature selection for clustering symbolic data. *Intelligent Data Analysis*, 4:19–28, 2000.

[67] D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York, 1985.

[68] V. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci.*, 98:5116–5121, 2001.

[69] Shivakumar Vaithyanathan and Byron Dom. Generalized model selection for unsupervised learning in high dimensions. In S. A. Solla, T. K. Leen, and K. R. Muller, editors, *Proceedings of Neural Information Processing Systems*, pages 970–976. MIT Press, 1999.

[70] A. B. van't Wout, G. K. Lehrma, S. A. Mikheeva, G. C. O'Keeffe, M. G. Katze, R. E. Bumgarner, G. K. Geiss, and J. I. Mullins. Cellular gene expression upon human immunodeficiency virus type 1 infection of CD4+–T–Cell lines. *J. Virol.*, 77:1392–1402, 2003.

[71] J. H. Wolfe. Object cluster analysis of social areas. Master's thesis, University of California, Berkeley, 1963.

[72] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17:763–774, 2001.

VITA

Nema Dean was born on November 9, 1979, in Dublin, Ireland. In 2002 she received her Bachelor of Arts (Honours) in Mathematics from Trinity College Dublin in Ireland. She graduated with a Doctor of Philosophy in Statistics from the University of Washington in 2006. In September 2006 she starts as a Lecturer in the University of Glasgow.