Fast High-Dimensional Bayesian Classification and Clustering

THÈSE N^{O} 4482 (2009)

PRÉSENTÉE LE 21 AOÛT 2009

À LA FACULTE SCIENCES DE BASE
CHAIRE DE STATISTIQUE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Vahid Partovi Nia

acceptée sur proposition du jury:

Prof. T. Mountford, président du jury
Prof. A. C. Davison, directeur de thèse
Prof. S. Morgenthaler, rapporteur
Prof. A. Murua, rapporteur
Prof. C.C. Taylor, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2009

Abstract

We introduce a fast approach to classification and clustering applicable to high-dimensional continuous data, based on Bayesian mixture models for which explicit computations are available. This permits us to treat classification and clustering in a single framework, and allows calculation of unobserved class probability. The new classifier is robust to adding noise variables as a drawback of the built-in spike-and-slab structure of the proposed Bayesian model. The usefulness of classification using our method is shown on metabololomic example, and on the Iris data with and without noise variables.

Agglomerative hierarchical clustering is used to construct a dendrogram based on the posterior probabilities of particular partitions, to provide a dendrogram with a probabilistic interpretation. An extension to variable selection is proposed which summarises the importance of variables for classification or clustering and has probabilistic interpretation. Having a simple model provides estimation of the model parameters using maximum likelihood and therefore yields a fully automatic algorithm. The new clustering method is applied to metabolomic, microarray, and image data and is studied using simulated data motivated by real datasets. The computational difficulties of the new approach are discussed, solutions for algorithm acceleration are proposed, and the written computer code is briefly analysed.

Simulations shows that the quality of the estimated model parameters depends on the parametric distribution assumed for effects, but after fixing the model parameters to reasonable values, the distribution of the effects influences clustering very little. Simulations confirms that the clustering algorithm and the proposed variable selection method is reliable when the model assumptions are wrong.

The new approach is compared with the popular Bayesian clustering alternative, MCLUST, fitted on the principal components using two loss functions in which our proposed approach is found to be more efficient in almost every situation.

Keywords: Classification; Clustering; Discrimination; Empirical Bayes; Hierarchical partitioning; Laplace distribution; MCLUST; Mixture model; Spike-and-slab model.

Résume

Nous proposons une approche performante pour la classification et le partitionnement pour des données hautement dimensionnelles continues, en se basant sur une mixture de modèles Bayésiens pour lesquels les formes analytiques sont accessibles. Ceci nous autorise alors de traiter le problème de la classification et du partitionnement simultanément ainsi que le calcul de probabilités pour des classes jusqu'alors non observées. La méthodologie développée a l'avantage d'être robuste à l'ajout de variables bruitées comme une conséquence de la structure "spike-and-slab" inhérente au modèle Bayésien proposé. L'utilité de notre modèle pour le problème de la classification est établie sur un exemple métabolomique ainsi que l'exemple classique des données "Iris", le cas de variables bruitées étant considérées ou non.

Le partitionnement hiérarchique ascendant est utilisé afin de construire un dendogramme à l'aide des probabilités *a posteriori* d'une partition donnée; ceci afin de produire un dendogramme ayant une interprétation probabiliste. Une généralisation au problème de sélection de variables est également proposée résumant l'importance des variables explicatives pour la classification et le partitionnement, tout en conservant son sens probabiliste. L'utilisation de modèles simples permet l'utilisation de l'estimateur du maximum de vraisemblance et mène ainsi à un algorithme entièrement automatique. La nouvelle approche de partitionnement est appliquée aux données métabolomique, "micro-array" et d'images ainsi que sur des données simulées motivées par des applications réelles. La complexité algorithmique de notre approche est discutée, des solutions afin d'en augmenter sa performance son alors proposées et le code développé est analysé.

Nos simulations montrent que la qualité d'estimation des paramètres du modèle dépend essentiellement sur l'hypothèse faite sur les distributions paramétriques affiliées aux effets. Cependant, après avoir raisonnablement fixé les paramètres du modèle, le choix de la distribution des effets joue alors un rôle nettement moins influent. Nos simulations confirment également que l'algorithme de partitionnement et la méthode de sélection de variables proposés sont pertinents lorsque les hypothèses faites sur le modèle sont fausses.

Notre approche est enfin comparée aux approches de partitionnement faisant référence, i.e. MCLUST, ajusté sur les composantes principales à l'aide de deux fonctions pertes. Ceci nous a permis de montrer que notre méthodologie était plus efficace dans la grande majorité des cas.

Mots-clés: Classification; Partitionnement; Discrimination; Bayes empirique; Partitionnement hiérarchique; Distribution de Laplace; MCLUST; Modèle de mixture; Modèle "Spike-and-slab".

Acknowledgement

I would like to thank my supervisor Anthony Davison for his generosity and for providing the opportunity to do research in his chair and for continuously directing the research and correcting my writings. He is a very encouraging supervisor both from a humanitarian and a scientific view and was always available to discuss any problems I encountered. I will never be able to thank him in a convenient manner that he truly deserves for the light he shed on my way which will guide me in the life passage forever. I would like to thank his family too.

I would like to thank Dr. Gaelle Messerli and Prof. Samuel Zeeman our biologist colleges who motivated this thesis by various insightful discussions. Also because they have provided the metabolite data.

I would like to thank my father, Hassan, my mother, Seddigheh, my little sister, Elaheh, for their comprehension and support during my whole life. Their effect was a bit hidden after moving to Switzerland but played a major role in settling me in my new life abroad.

I would like to thank my older sister, Raheleh, for joining me in Switzerland after a year of starting this thesis and her explanations about GC-MS technology and providing Figure 1.1. I would like to thank my brother in law, Reza, for spending unforgettable time together.

I would like to thank Sierro family in particular Jean-Michel and Elisabeth for becoming my family in Switzerland and especially Nicole for sharing both happy and difficult moments during the recent three years.

I would like to thank Kjell Konis for his help with building the `R` packages related to my thesis. I would like also to thank Arpit Chaudhary.

I would like to thank Anne-Lise Courvoisier, the secretary of the chair of Statistics, for managing administrative works and for her considerable help with Swiss official procedures when I entered in Switzerland with no knowledge in French language.

I would like to thank Prof. David Stephens from the McGill university and Prof. Chris Holmes from the Oxford university for their kind letter of support, also the Swiss National Science Foundation for providing me a joint fellowship to continue doing research on this subject.

Contents

1 Preliminaries **11**

1.1 Introduction 11

1.2 Biological Background 13

1.2.1 Generalities 13

1.2.2 Metabolomics 13

1.2.3 Microarrays 15

1.3 Data Examples 15

1.3.1 Metabolite Data 15

1.3.2 Microarray Data 17

1.4 Data Exploration 17

1.4.1 Basics 17

1.4.2 Metabolite Data 20

1.4.3 Microarray Data 21

1.5 Purpose of Thesis 21

2 Classification **25**

2.1 Introduction 25

2.1.1 General 25

2.1.2 High-Dimensional Classification 29

2.2 Hierarchical Bayesian Classification 31

2.2.1 General 31

2.2.2 Gaussian Effects Model 35

2.2.3 Asymmetric Laplace Effects Model 38

2.3 Extensions 41

2.3.1 Unobserved Class Probability 41

2.3.2 Built-in Variable Selection 42
2.4 Examples 46
2.4.1 Metabolite Data 46
2.4.2 Iris Data 56
2.5 Analytical Calculations 61
2.5.1 Introduction 61
2.5.2 Joint Density 61
2.5.3 Density for Inactive Variables 62
2.5.4 Density for Active Variables 63
2.5.5 Multivariate Gaussian Density-Distribution Integral . . 68
2.5.6 Likelihood 69

3 Clustering **71**
3.1 Introduction 71
3.1.1 General 71
3.1.2 Model-Based Clustering 75
3.1.3 High-Dimensional Clustering 77
3.1.4 Clustering Prior 79
3.2 Hierarchical Bayesian Clustering 81
3.3 Computational Issues 87
3.3.1 General 87
3.3.2 Joint Density Acceleration 87
3.3.3 Individual Density Computation 87
3.3.4 Density of the Gaussian Model 88
3.3.5 Density of the Asymmetric Laplace Model 90
3.3.6 Code Analysis 91
3.4 Examples 93
3.4.1 Metabolite Data 93
3.4.2 Microarray Data 105
3.4.3 Image Data 108
3.5 Discussion 109

4 Simulation Results **115**
4.1 Introduction 115

4.2 Gaussian Effects Model 119
4.3 Asymmetric Laplace Effects Model 130
4.4 Parameter Estimation 140
4.5 Heavy-tailed errors 150
4.6 Correlated Observations 154
4.7 Summary 161

5 Conclusion and Discussion **163**

Chapter 1

Preliminaries

1.1 Introduction

In the Oxford dictionary a *cluster* is defined as "a group of similar things positioned or occurring closely together". In physics a cluster is a small group of atoms and molecules, in computing a group of coupled computers that works together, and in system file management a group of disk sectors used in the file allocation system.

The composition of *cluster* with other words makes the range of its meanings even wider, especially in technical language. For instance, a *cluster-headache* is a neurological disease in medicine, a *cancer-cluster*, in biomedicine, is a greater-than-expected number of cancer cases, and *cluster sampling* is a random sampling method in survey sampling.

Cluster analysis or *clustering* refers to methods of grouping similar subjects. It is one of the most fundamental and correspondingly the most useful learning techniques. Cluster analysis facilitates the study of complex systems by putting similar items in a group, because it is usually easier to find governing rules by looking at similar objects.

Before the modern era the world was studied mostly qualitatively, and qualitative clustering was used to extract knowledge from the universe. For example, Aristotle divided living organisms into two groups, plants and animals, and divided animals into three groups according to how they moved: walking, flying, or swimming. Avicenna used observational characteristics of

patients to try to describe responses to drugs by classifying them into four groups. Galileo used the similarity of movement of groups of stars and discovered that they are positioned in the same galaxy, which today is called the Milky Way. Mendeleev discovered the periodic table by grouping elements having similar chemical characteristics. Darwin studied the evolution of species by clustering biological bodies.

The development of measurement systems led to the possibility of quantitative clustering. The wide application of clustering in different domains has made its literature large and diverse, with many synonyms in various fields: cluster analysis in statistics, numerical taxonomy in biology, pattern recognition in engineering, unsupervised learning in machine learning, clumping in linguistics, regionalisation in geography, and partitioning in graph theory (Anderberg, 1973). Immediately after the fundamental development of clustering ideas, textbooks were authored and published, which also shows their importance from the early stages of their appearance; classic texts include Tryon (1939) and Sokal and Sneath (1963). After the introduction of computers, books were reauthored and modified (Tryon and Bailey, 1970) and rapid advances in computational power demanded continuous update (Kaufman and Rousseeuw, 1990; Gordon, 1999; Everitt *et al.*, 2001; Abonyi and Feil, 2007).

As described earlier, clustering divides the observations into homogeneous groups. Once this has been done a new observation may arrive. Assignment of the new object to one of the already observed groups is usually called classification, discrimination, supervised pattern recognition, or supervised learning. Classification in statistics is a method of finding the missing label of a new observation based on a training set of labelled data. Hence there is a close relationship between classification and clustering: when there is no label observed, and estimation of labels of the whole data is of interest, it is called clustering, but when labels are already observed and estimation of the label of a new observation is required, it is called classification.

1.2 Biological Background

1.2.1 Generalities

Many technologies today provide signals and data with experimental and measurement errors. In almost all subdomains of biology, such data are used to confirm or deny scientific assertions based on a statistical model. The common difficulty of analysing biological data is to provide a valid statistical model for low-sample-size-high-dimensional situations. Due to recent technological advances, the precision of measuring the chemical composition of biological bodies has increased, so the number of variables recorded has augmented to give high-dimensional data; on the other hand, due to time and budget constraints, the number of tissues or patients under study is limited, which yields low sample sizes. Such data are statistically troublesome, because in classical statistics, properties of methods are studied asymptotically—that is, when the sample size tends to infinity—which apparently is not practical for modern data settings.

1.2.2 Metabolomics

Biologists continuously discover the ability of biological systems to isolate harmful processes and improve their behaviour. Recent advances have guided scientists to focus on studying the genetic make-up of organisms, but this gives only part of the information required to observe its response to stimuli. For a more comprehensive view, it is also required to investigate the dynamical change of biomarkers, the real-time signals which reflect the integrated functionality of the organism.

Metabolites are intermediate or end products of metabolism, and the term *metabolom* commonly refers to the set of metabolites found in a cell. *Metabolomics* is defined as the quantitative measurement of metabolic response of living systems to physiological or genetic modifications, measured by means of analytical chemistry tools. It is a challenging domain which is used to make deductions about the functionality of metabolites and their usefulness in fingerprinting biological tissues. There is a close relationship between metabolomics, systems biology, proteomics and genomics, which all

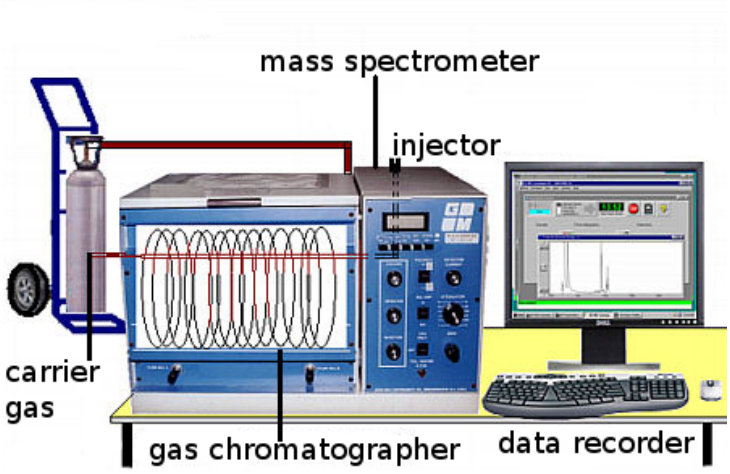

Figure 1.1: The Gas-Chromatography-Mass-Spectrometry instrument drawn based on the Gas-Chromatography-Mass-Spectrometry Wikipedia article (`http://wikipedia.org`).

try to give a complete picture of living organisms.

One of the most-used technologies in metabolomics is Gas-Chromatography-Mass-Spectrometry (GC-MS), which helps to quantify metabolites of tissues. The GC-MS separation method has had great technological and industrial success since its introduction about fifty years ago (Gohlke, 1959) and currently is applied in a wide range of fields, including chemistry, biotechnology, medicine and food industry. The GC-MS instrument consist of integrated subsystems performing four steps (Gohlke and McLafferty, 1993): vaporisation and chromatic separation; ionisation of the sample vapour; mass separation of the ions; and amplification and recording of the detected signals. Because of the vaporisation step, the reference compounds and the unknown compounds should be thermally stable. First, the reference solution is injected into the GC inlet, where it is vapourised and passed to the separation column by the carrier gas. Then, the separated compounds pass through a heat transfer line and are ionised. Finally, the mass-to-charge ratio is measured by a detector and recorded in a computer, see Figure 1.1.

1.2.3 Microarrays

Genetic science has revolutionised our daily life. Biologists have discovered the role of many genes in the biological process and altered them toward our needs. For example genetics is used to produce certain substances, many of them expensive to obtain by artificial manufacturing methods. Scientists can alter bacteria so that they produce specific proteins, like the insulin needed for patients with diabetes. The genetic configuration of cows is altered to produce more milk and force them to grow faster. In medicine, genetic information may be used to choose the best therapy for a patient.

Every cell of the body contains a full set of chromosomes and identical genes, but only a fraction of these genes are turned on (expressed). It is the expressed subset which reflects the unique properties of each cell type. The genetic information of a cell is coded in chromosomes which are located in the nucleus of cell and consists of deoxyribonucleic acid (DNA) sequences. The messenger ribonucleic acid (mRNA) is transcripted from a DNA template and carries the encoded genetic information.

Extraction of genetic information from body cells is complex. The technology typically used is an *mRNA microarray*, which consists of series of micro spots arranged on a microscope slide, each one containing a different mRNA sequence. A fluorescent RNA sample is hybridised to the slide and then the intensity of the fluorescent in each spot is measured and analysed. It is common to hybridise two samples at the same time, a test sample labelled with red fluorescent *Cy3*, and a control sample labelled with green fluorescent *Cy5*. Then, the ratio of the intensity of *Cy3* to *Cy5* is measured by a sensor and stored for future analysis.

1.3 Data Examples

1.3.1 Metabolite Data

The metabolite data consist of 14 genetically modified samples of the plant *Arabidopsis thaliana* grown in three batches. Values of 43 metabolites are measured for each sample which are supposed to monitor their genetic changes. The data involve two mutants defective in starch biosynthesis, *pgm* and *isa2*;

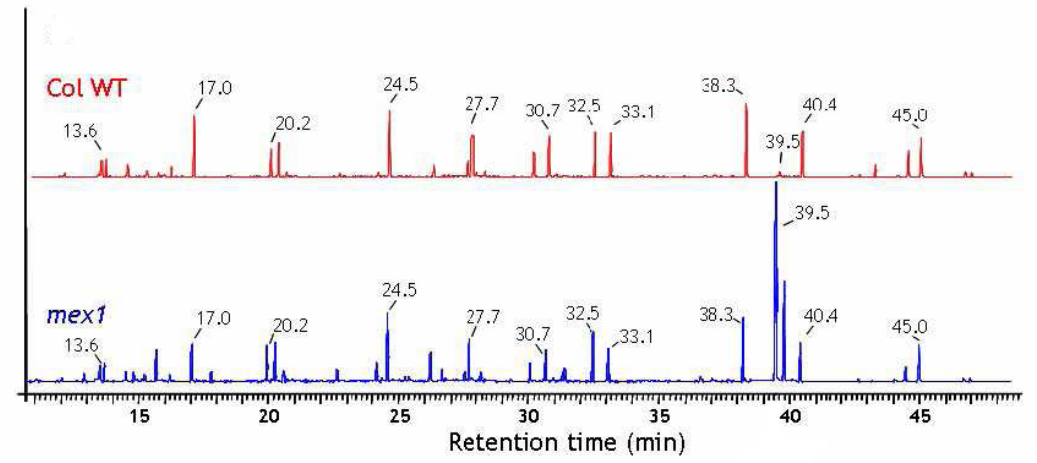

Figure 1.2: The Gas Chromatography/Mass Spectrometry spectra of a replicate of *ColWT* (top spectrum) and a replicate of *mex1* (bottom spectrum), from Messerli (2007). The horizontal axis shows the time (minutes) at which the peaks are observed.

four defective in starch degradation *sex1*, *sex4*, *mex1*, and *dpe2*; a mutant for comparison that accumulates starch as a pleitropic effect, *tpt*; four uncharacterised mutants, *deg172*, *deg263*, *ke103*, and *sex3*; and three wild type plants, *WsWT*, *RLDWT*, and *ColWT*. There are four replicates of all samples except the last for which there are three (Messerli *et al.*, 2007).

GC-MS technology is used to measure the metabolites providing a cheap and fast method to extract a large number of relatively low molecular-weight metabolites. The sample GC-MS spectra of replicates of *ColWT* and *mex1* are shown in Figure 1.2.

The mutants are grown in three batches, the amount of the metabolites' content measured by the GC-MS technology and, finally the data are normalised with respect to a control mutant included in each batch to help eliminate the batch effect. The purpose of the study was to identify the uncharacterised mutants and describe biologically the metabolic similarity of different classes.

The log-transformed data are shown in Figure 1.3. Each line corresponds to a replicate of a mutant represented across metabolites. The metabolites are ordered with respect to their variance across samples. The variance of

each metabolite is calculated and then the first metabolite is the metabolite having the maximum variance, the second metabolite is the metabolite having the second biggest variance and so on. The variance of a metabolite is somehow a measure of its importance, because variables which take constant or nearly-constant values cannot characterise samples and hence are useless for classification or clustering. The profile plot in Figure 1.3 shows that two mutants *mex1* and *dpe2* behave differently on variable *maltose.MX*. The wild types *ColWT*, *RLDWT*, *WsWT* all have a nearly flat profile and the pattern of the other mutants is not very clear.

1.3.2 Microarray Data

The microarray data consist of gene transcripts of patients with brain cancer. The data are available through the *GEO* website, which is built to make gene expression data available online and free of charge. The website includes a rich gene-expression data repository accompanied with basic statistical analysis tools; see `http://www.ncbi.nlm.nih.gov/geo/`. The dataset `GDS1975` involves 74 brain tumours measured on $22,000$ genes to study wheather the gene expression of the brain tumour is predictive of survival of the patients having brain cancer. Freije *et al.* (2004) select a subset of 595 genes in their analysis, but even after correspondence with the authors, we were able to match only 396 of 595 genes they used. The data are normalised in two steps, first by taking the natural logarithm, and second by subtracting the median of each gene. An image plot of the data is shown in Figure 1.4.

1.4 Data Exploration

1.4.1 Basics

Understanding basic properties of data, or data exploration, plays an important role in building a successful statistical model. It helps toward having a general idea for formulating the parameters of interest, and also may assist when choosing an appropriate family of distributions.

Visualising high-dimensional data is burdensome, because of the restric-

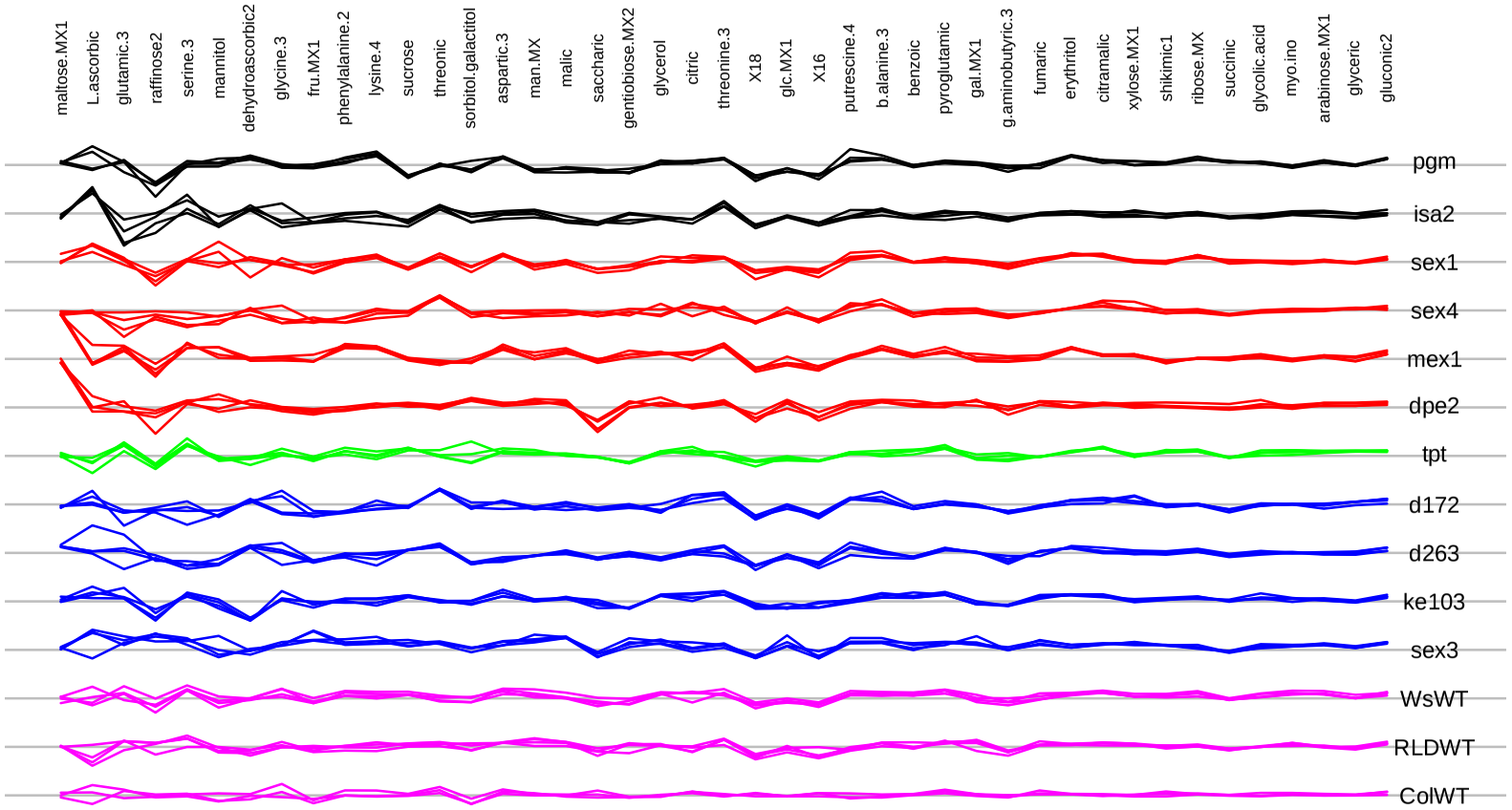

Figure 1.3: A profile plot of the metabolite data. The samples across metabolites are plotted using solid lines. The metabolites are ordered with respect to their variance. Different colours are used to represent the mutants' category: black for the plants defective in starch biosynthesis, red for those defective in starch degradation, green for the comparative plant *tpt*, blue for the uncharacterised mutants, and magenta for the wild types.

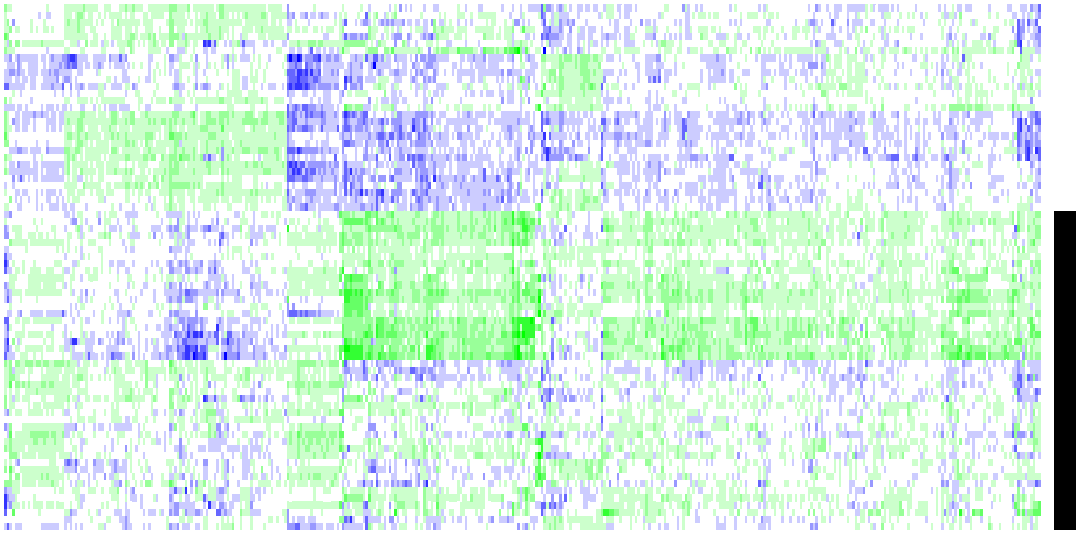

Figure 1.4: An image plot of the microarray data, white is used when the gene in that sample is not expressed, green if intensity of the test sample is less than control, and blue otherwise. The survival time of patients is categorised into two groups. Survival group 1 is represented in white (29 patients), and survival group 2 is shown in black (45 patients) (Freije *et al.*, 2004). See also Figure 1.7.

tion of the visual imagination to three dimensions. In such cases projection on subdimensions is usually used or important dimensions are selected to visualise the data on.

Principal component analysis is one of the most-used projection techniques for high-dimensional data. The idea is to find a set of linear and perpendicular transformations with largest variance (Johnson and Wichern, 2007). The coefficients of the linear transformation are called *loadings* and the projected data are called *scores*. The estimation of the variance-covariance matrix of the data plays the key role in calculation of the principal axes. However, it becomes non-singular when the number of dimensions exceeds the sample size. Hence, pre-selection of variables is required for implementation of principal component analysis on high-dimensional data.

A dimension that does not vary across samples cannot characterise them and hence is useless for grouping. On the other hand, such variables take small loadings in principal components analysis. Hence, loadings may give

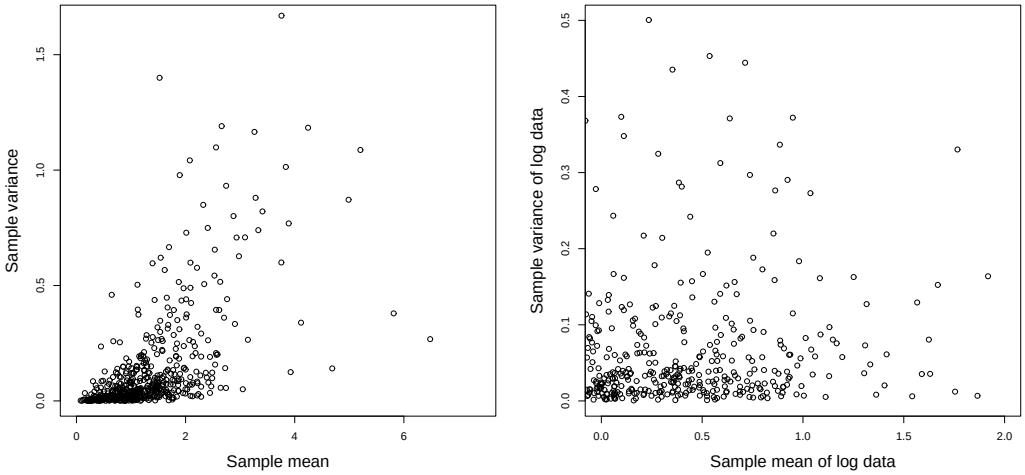

Figure 1.5: The sample means and variances of four replicates of each mutant for each variable. For example, two mutants with R replicates measured each on ten variables produce 20 points. The left panel is for the original data and the right panel is the right panel is for the log data.

restricted information about the importance of variables for classification or clustering.

The principal component analysis is implemented on n variables having largest variance across n samples.

1.4.2 Metabolite Data

The Gaussian distribution is widely used to model continuous data. In the Gaussian class, the sample mean is independent of the sample variance, which apparently is not true in our data, see the left panel of Figure 1.5. Hence, we may apply the Box–Cox transformation (Box and Cox, 1964), in our case the log transformation, to make the relationship between the mean and variance disappear.

Principal component analysis is implemented twice, once on the mean of the replicates of each plant (Figure 1.6, top panel) and once on the whole sample (Figure 1.6, bottom panel). Comparing the result of this analysis with the plant profiles (Figure 1.3) confirms that variables with large variance, in

general, have large principal component loadings; for example compare the metabolites *maltose.MX, L.ascorbic, glumatic.3, raffinose2* in Figures 1.3 and 1.6.

After projection of the mean of the replicates onto the two principal component axes, three groups are visible: a group with nearly flat profiles, that is all wild types, *ke103, sex3*, and *tpt*; another group comprising the two mutants that behave differently for variable *maltose.MX*: the mutants *dpe2* and *mex1*; the last group involving the remaining plants *isa2, sex4, d172, d263, pgm,* and *sex1*. Looking at the whole samples projected on the two principal component axes, only two groups become visible: the group of samples of *mex1* and *dpe2*, and another group including the remaining samples. Apparently, disregarding information related to replication may change the grouping, and so lead to a different classification or clustering.

1.4.3 Microarray Data

In analysis of microarray data, principal component analysis is frequently used as a dimension reduction method for visualisation and preprocessing (Alter *et al.*, 2000; Li and Li, 2004). Like the metabolite data we apply principal components; and because the number of variables are more than the sample size we select the variables with greatest variance to implement the method. The number of variables for the analysis is chosen to be 74 out of 396, to ensure non-singular estimation of the variance-covariance matrix. The data projected onto the two principal axes are shown in Figure 1.7, which indicates no clear grouping. Also, it appears that survival groups are barely separable, especially for points projected close to the origin. This confirms that correct classification of such points is troublesome.

1.5 Purpose of Thesis

The metabolite data has motivated the work of this thesis, whose the main objective is to answer the following questions

- The metabolite behaviour of the unknown types *d172, d263, sex3,* and *ke103* is closest to which of the known types?

- How similar are these unknown plants to the known types?
- Is it possible that mutants come from a category that is not in the known types provided in the training set?
- Which metabolites characterise the known and unknown types?
- Which known and unknown mutants follow similar metabolite patterns?

We answer the first four questions in Chapter 2 and the last question in Chapter 3.

The microarray data are used to test the clustering method of Chapter 3 on a higher-dimensional dataset.

This research presents Bayesian parametric models for classification and clustering of high-dimensional data which to our knowledge has not been considered before. The proposed models have closed-form joint densities which are used to establish a fast classification and clustering algorithm, and to provide estimation of model parameters using maximum likelihood.

In high-dimensional data, usually, a small subset of variables is informative and considering unnecessary dimensions yields poor results due to overfitting. In order to make the result less affected by noise variables, we propose a built-in spike-and-slab structure, which helps to quantify the importance of variables.

Our proposed approach solves classification and clustering in a single framework. Agglomerative clustering is implemented to provide a visual grouping of subjects by a dendrogram, and the log-posterior is proposed as a natural distance imposed by the model to construct the tree. The posterior-based dendrograms have a probability interpretation, helping to suggest which grouping is more likely under the model. The optimal grouping is found by cutting the dendrogram using the maximum-a-posteriori principle, yielding a fully-automatic and fast clustering method. Our simulation studies confirm the good performance of our proposed procedures compared with *MCLUST*, a popular automatic Bayesian clustering algorithm.

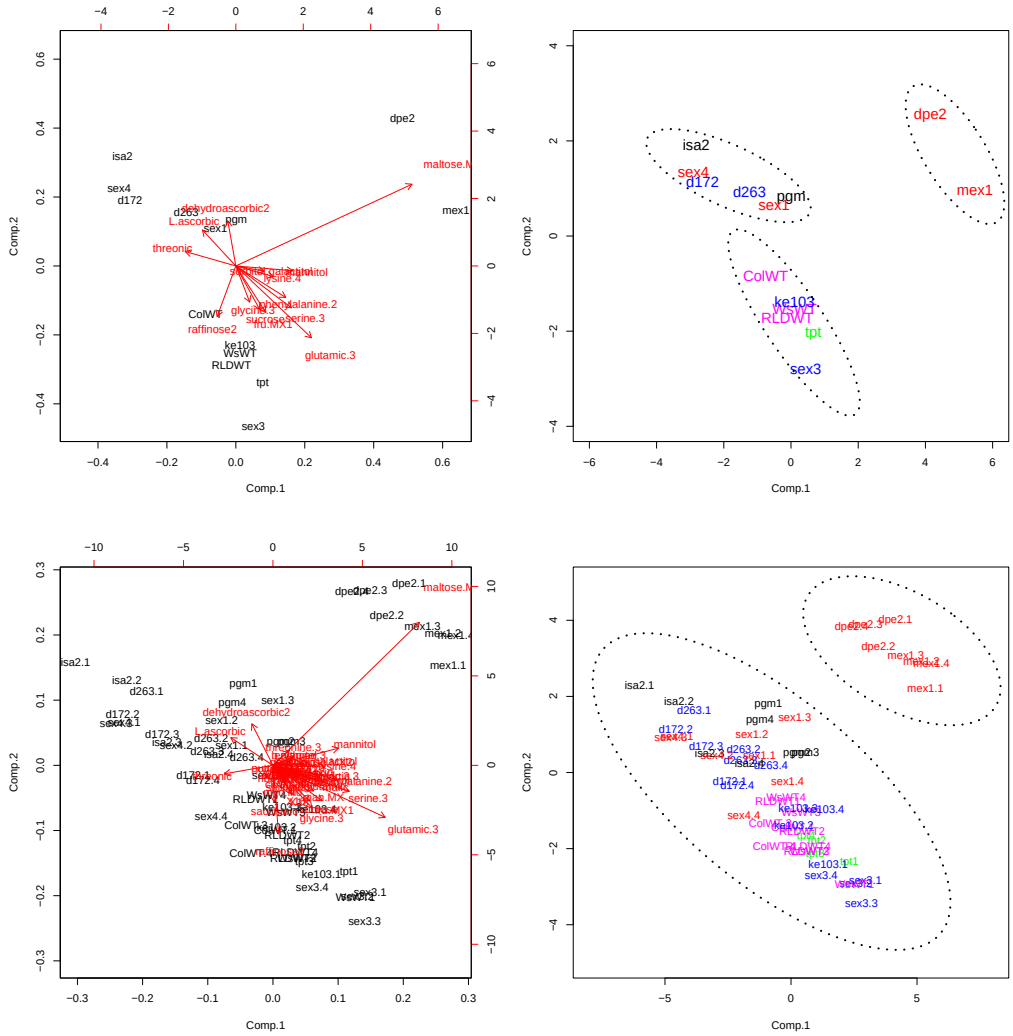

Figure 1.6: The mutants mean profile (top figures) and all samples (bottom figures) of the metabolite data projected on two principal components. The overlaid plot of scores and loadings is represented in the left side, with visual grouping of scores highlighted by dashed ellipses in the right side, respecting the colour scheme of Figure 1.3.

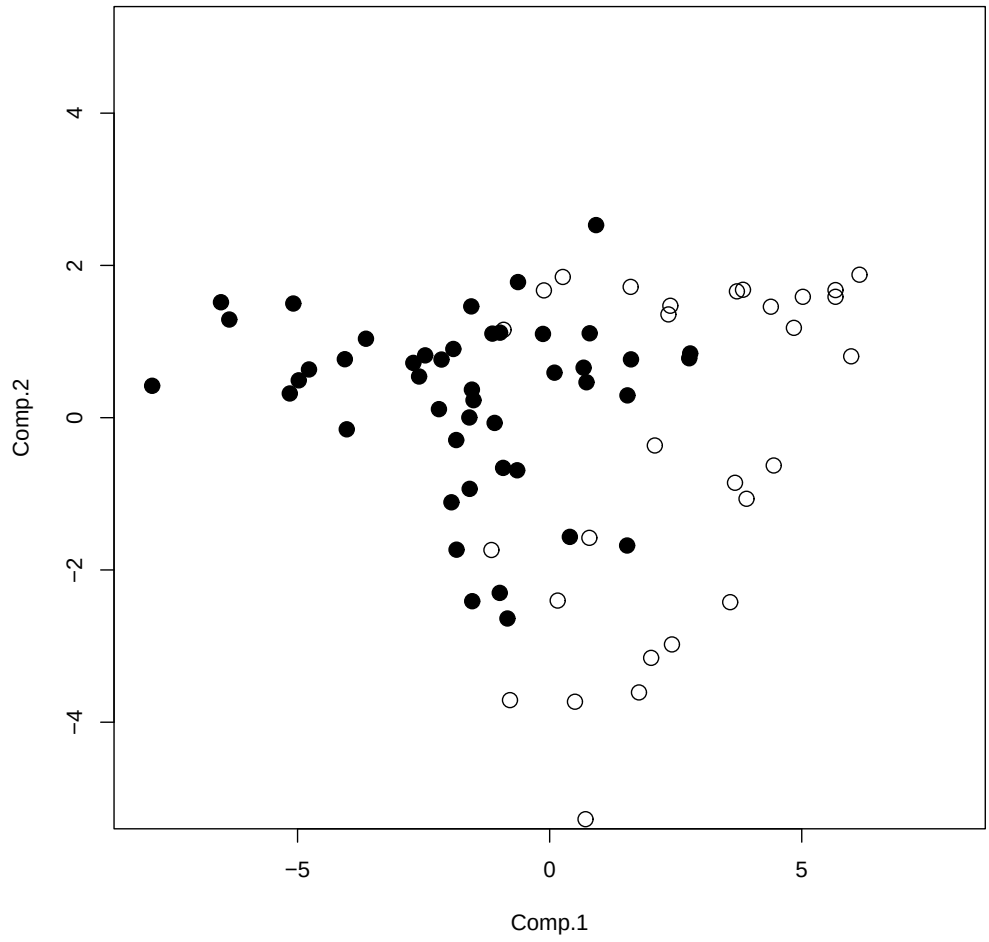

Figure 1.7: The microarray data projected on two principal components. The samples with group 1 are represented by white circles and observations with group 2 are shown using solid black blobs; see also Figure 1.4.

Chapter 2

Classification

2.1 Introduction

2.1.1 General

Classification and discrimination are multivariate techniques concerned with separating sets of objects and allocating new observations to previously defined groups. They are implemented to achieve two main goals. The first is to describe classifying features of objects using collections of features hidden in measured variables. The second is to derive a rule that can be used to optimally assign new observations to previously defined classes, see Figure 2.1. The word discrimination was used by R. A. Fisher (Fisher, 1936) to express the first goal and classification or allocation often refers to the second goal. Classification methods can be grouped into two categories: parametric (model-based), and nonparametric (model-free). In a parametric method a probability model is assumed for the classes and the new object is classified to the class having the largest density at the observed point. A nonparametric method attempts to estimate the classification borders using a flexible smooth function and the new observation is attributed according to the estimated border, but sometimes a distance is defined, and the new individual is assigned to the closest class using the defined distance; we call this a distance-based method. However, it is hard to clearly distinguish nonparametric from parametric techniques. For instance a distance may be defined in a way that

corresponds to a density (or log density) at the observed point, and hence coincides with a parametric approach. The difference between model-based or model-free methods is the direct assumption of a data distribution, but they may yield the same classification rule.

Fisher's linear discriminant axes, for groups with the same covariance matrix, seek uncorrelated linear combinations of variables that separate the data as much as possible. Separability is defined using the ratio of between-class to within-class variance. Fisher's discriminant axes are useful in interpreting features of the data and visualising them. Each axis gives a classification rule and often the first axis is used to classify a new subject.

A discriminant is a multivariate function of the measured variables and hence may geometrically be regarded as a hyperplane. If observations are separable by a linear hyperplane, the separating hyperplane may not be Fisher's linear discriminant. In machine learning, separating data by a hyperplane is studied to create the classification rule and often is called a perceptron (Rosenblatt, 1958). This later became a basis for the *neural network* classifier (Hastie *et al.*, 1995). If the data are separable by a linear hyperplane, then its existence ensures an infinity of such hyperplanes. Vapnik (1996) maximised the distance between the plane and the nearest points, called margins, and made the solution unique because just one of the infinite linear hyperplanes has the maximum margin from its nearest data points. Vapnik's idea was later developed to generalise the technique for cases where data are not separable, yielding *support vector machines* (Hastie *et al.*, 2001, Chapter 12). Nonlinear classification rules can be handled similarly by replacing the linear hyperplane with a flexible kernel. Empirical evidence suggests that support vector machines perform well in learning problems. However, simplicity, having a closed form solution, and being computationally efficient make Fisher's linear discriminant a widely used classification method. Mathematical arguments support the linear discriminant as an optimal method according to zero-one loss under some specific choices of distribution. The zero-one loss equals zero if the new observation is correctly classified and equals one otherwise. The optimal method according to the zero-one loss, is the linear discriminant for multivariate Gaussian data with common covariance matrix (Johnson and Wichern, 2007, pp. 579–584).

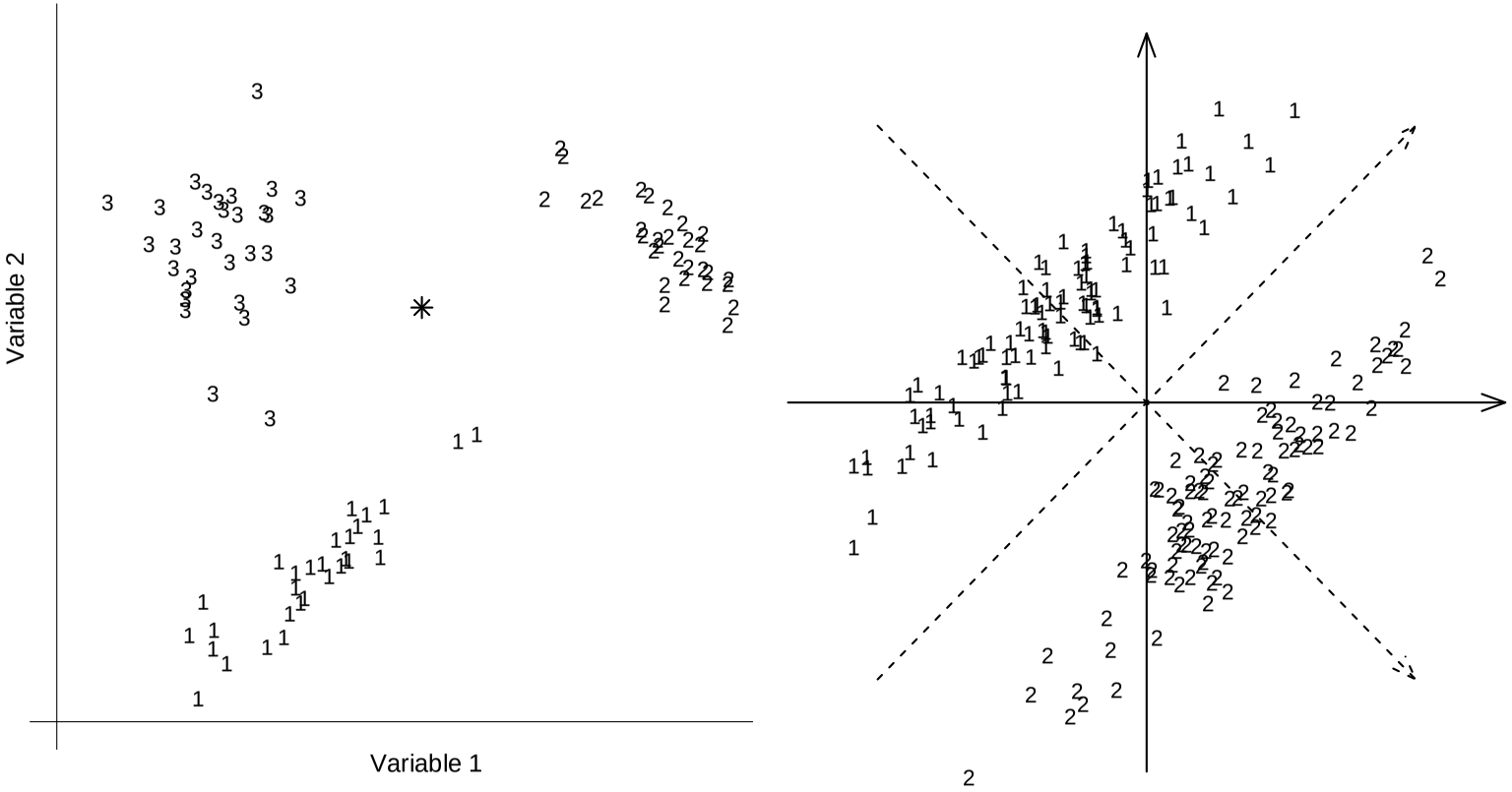

Figure 2.1: The classification problem shown on data with three classes and two variables; a new observation (*) must be classified to one of the observed groups (left panel). The original variables (solid axes) and the linear discriminant axes (dashed axes) are shown for data with two groups (right panel).

In high-dimensional situations it is hard to propose a practically useful multivariate distribution because distributions in high dimensional situation involves often a lot of parameters, which can be hard to estimate reliably using the available data. Often multivariate models with a small number of parameters work well in classification (Hand and Yu, 2001). However, applying distance-based methods is more common in high dimensions, nearest neighbour methods may be regarded as generalised versions of distance-based techniques; for a review see Hastie *et al.* (2001, Chapter 13). The advantage of distance-based methods is in providing a simple and understandable discriminant function. However, it is hard to prove their optimality as simply as for the model-based techniques. Intuitive distances such as Euclidian distance are inappropriate for classification; for discussion see Chan and Hall (2009).

Regression classifiers also fall in the model-based classification category. In regression classification, the response variable is the class indicator, a discrete random variable, and the other measured variables are explanatory. The explanatory variables sometimes are called classification variables too. The response variable is connected to the classification variables by a model, often a log-linear one. A regression classifier requires no assumption about the distribution of the explanatory (classification) variables, since the model is built using the conditional distribution of the response given the classification variables.

Sometimes different classification rules are proposed which are all individually weak. Improving the accuracy of weak classifiers using a specific procedure, called an *ensemble method*, has become a hot topic in recent decades. Ensemble methods in classification refer to weighting or to aggregating individually weak classifiers in order to build a powerful classification rule. Applying this by re-sampling is referred to as *bagging* (Breiman, 1996), and applying it by over-weighting misclassified observations is called *boosting* (Freund, 1995). Friedman *et al.* (2000) reformulated bagging and boosting in terms of additive logistic regression and showed that improvement in classification by ensemble methods is tightly connected to the maximum likelihood principle.

Sometimes the uncertainty of classification is of interest. In the model-

based classification uncertainty of classification can be obtained using Bayes' theorem. A prior probability for classification of the observation to each group is assumed, such as a uniform discrete distribution, and the posterior probabilities are calculated. The higher the posterior classification probability, the more certain the classification is.

2.1.2 High-Dimensional Classification

The main difficulty in high-dimensional data analysis is the *curse of dimensionality*, which may be regarded as stating that adding dimension without adding observations yields exponential increase of empty hypercubes (Bellman, 1961). In order to see the problem, assume a step function approximating an unknown function from which we have two observations. The unknown function can be regarded as the classification rule and the observations as the training data. Adding extra variables without adding observations yields two regions with no observation. In three dimensions, six regions with no observation exist, see Figure 2.2. Hence, the estimation of a specific function using the same number of observations becomes more difficult in higher dimensions.

In order to avoid the curse of dimensionality, dimension reduction or variable selection is required. A good reduced space for classification is given by the linear discriminant axes, since by definition these seek the linear projection in which the data are as separable as possible. The linear discriminant axes are optimal linear transformations if classes have a common covariance matrix, but this is hard to check in high-dimensional situations. Even if it is true, we may not have enough samples to estimate the mean vectors and the common covariance matrix reliably. For example, assuming a classification problem with C classes and multivariate Gaussian model with common covariance matrix for the data, full-rank estimation of the covariance matrix requires at least $V(V+1)/2 + VC$ observations, where V is the number of classification variables. Assuming unequal covariance matrices, at least $V(V+1)/2 + V$ observations are required for each class, giving $C(V^2 + 3V/2)$ observations in all. If the sample size is not moderately large, shrinkage toward a common covariance matrix has been proposed (Friedman,

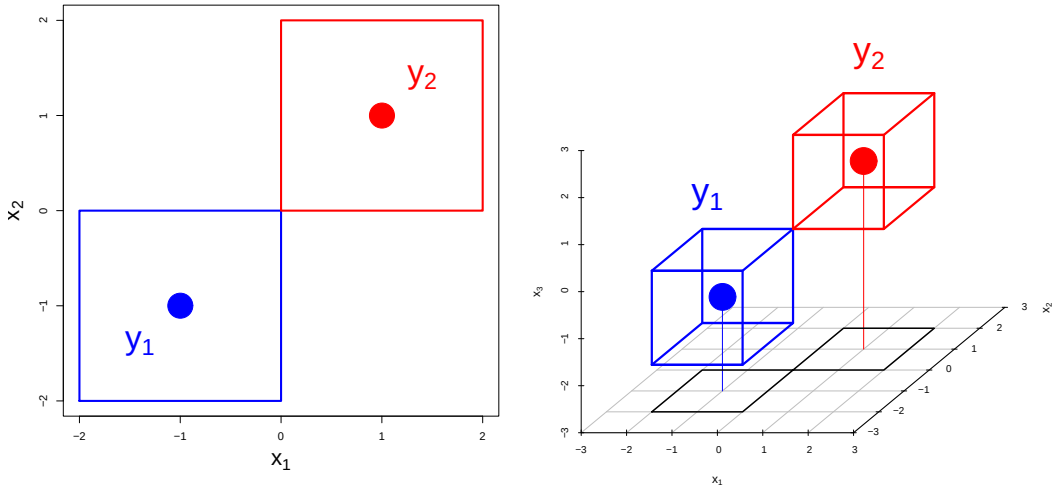

Figure 2.2: The curse of dimensionality: empty hypercubes increase exponentially with variables. Dimensions are denoted by x and the training data comprising two observations are denoted by y_1 and y_2 and shown using blobs.

1989). However, there are situations where even assuming a common covariance matrix still overfits the data. In order to avoid over-fitting, a structural version of the covariance matrix, with less parameters, may be useful too.

In regression classification, it is often assumed that the regression coefficients are sparse, and sparse estimates are obtained by penalising the likelihood with the L_1 norm (Park and Hastie, 2007). The availability of fast algorithms to compute the model parameters and select variables jointly has popularised the use of penalised likelihood methods.

Bayesian approaches to high-dimensional classification have been less considered, maybe because of the complexity of the resulting posterior and a lack of fast and reliable sampling tools. Bayesian variable selection typically requires trans-dimensional Markov chains, such as the reversible jump algorithm of Green (1995), to sample from the posterior distribution. Such Markov chains often have slow rates of convergence. Other methods require ordinary Markov chain Monte Carlo (George and McCulloch, 1997).

A variable selection has 2^V states, exponential in the number of variables. Thus in the Bayesian approach, if one likes to give the possibility of one visit

for each state, the chain should be run for at least 2^V iterations, which is large for high-dimensional cases. For instance for data with $V = 50$, 100, and 1000, the number of states is about $10^{15}, 10^{30}$, and 10^{300}, respectively.

In contrast, penalised regression alternatives like the lasso give sparse estimates and select variables by running a quadratic optimisation (Tibshirani, 1996). A faster algorithm for the lasso was proposed by Efron *et al.* (2004). Recently, the Dantzig selector was proposed, which requires only a linear optimisation (Candes and Tao, 2007), and its computationally efficient optimisation has been explained in James *et al.* (2009).

Spike-and-slab models are general methods for implementing variable selection in a Bayesian context (Mitchell and Beauchamp, 1988; Brown *et al.*, 1998), by assuming a mixture prior for effects, one component concentrated about zero and another with spread tails. In this thesis we present a Bayesian approach to classification using a spike-and-slab hierarchical model. In order to achieve a computationally fast method we propose models with analytically closed-form posteriors. As a consequence, we assume independent classification variables, which appears to be restrictive, but works well for small-sample-size-high-dimensional cases (Hand and Yu, 2001; Hand, 2006).

This chapter is organised as follows. Gaussian and asymmetric Laplace hierarchical models are introduced in Section 2.2 and classification using them is discussed. Extensions to classification of a new observation, which also allows calculation of the probability of being in a previously unobserved class is given in Section 2.3, and then a generalisation of the hierarchical models which allows variable selection is introduced. The novel methods are compared with the linear discriminant and naive Bayes approaches on the metabolite data and the famous iris data in Section 2.4. The details of calculations for the spike-and-slab models are given in Section 2.5.

2.2 Hierarchical Bayesian Classification

2.2.1 General

In this section we introduce a novel approach to high-dimensional classification by considering a Bayesian linear model with a spike-and-slab structure.

It is known that the classification probabilities in high dimensions degenerate and hence cannot be used as an uncertainly measure for allocation (Hand and Yu, 2001). Our models guide the classification function to allocate new observations according to important similarities and gives posterior classification probabilities which rarely degenerate. Hence using our suggested approach, the posterior probabilities can be interpreted as a measure of similarity of the new object to one of the classes. The proposed models have analytically tractable marginal posteriors, and hence the classification probabilities are rapidly calculated. In addition, the model parameters can be estimated using maximum likelihood.

The measured quantities to be used for classification are modelled as follows. Assume that the univariate random variable y_{vctr} is the rth ($r = 1, \ldots, R_{ct}$) replicate of type t ($t = 1, \ldots, T_c$) from class c ($c = 1, \ldots, C$) measured on variable v ($v = 1, \ldots, V$). Thus V continuous variables are measured on C classes, each class consisting of T_c types of individual with R_{ct} replicates of the tth type. Hence the total number of types is $T = \sum_{c=1}^{C} T_c$ and the total number of observations for each variable is $\sum_{c=1}^{C} \sum_{t=1}^{T_c} R_{ct}$. In some cases the measurements are unreplicated, and then $R_{ct} = 1$ for $t = 1, \ldots, T_c$ and $c = 1, \ldots, C$. The result of the measurement is a scalar y_{vctr}, which we assume may be expressed as

$$y_{vctr} = \mu + \gamma_{vc}\theta_{vc} + \eta_{vct} + \varepsilon_{vctr}, \tag{2.1}$$

where θ_{vc}, η_{vct} and ε_{vctr} are independent continuous random variables, and γ_{vc} is a binary random variable with success probability p. In equation (2.1), μ represents an overall value for all the variables and types. Without loss of generality, our model presupposes that the data have been modified so that the variable-wise averages equal zero, but that the variances have been left unaltered. If $\gamma_{vc} = 1$, then this variable-class combination is active, and then the profile in an ideal setting would be $\mu + \theta_{vc}$. If $\gamma_{vc} = 0$, then the combination is inactive and the profile in an ideal setting would be μ. No realisable setting is ideal, however, and additional variation between types—for example, due to varying experimental conditions—is represented by the variables η_{vct}. Finally the lowest level of variability between replicates is due to measurement error, ε_{vctr}.

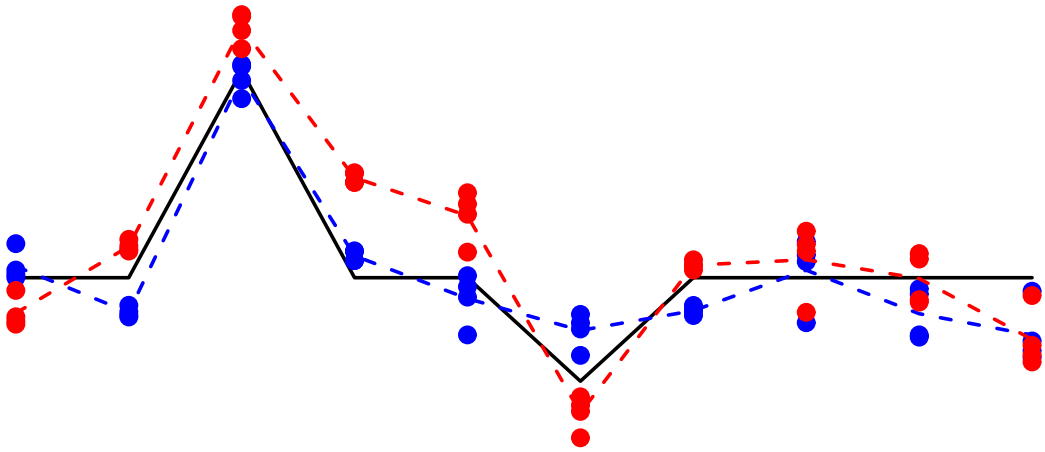

Figure 2.3: Graphical form of model (2.1), showing the true effects (solid black line), and two observed profiles (blue and red dashed lines) of a class each with four replicates (solid blobs).

Figure 2.3 shows the model in graphical form. If the variables measured from a single type are visualised as a profile, then the sharp solid ideal profile $\mu + \gamma_{vc}\theta_{vc}$ corresponding to class c is blurred to the dashed lines of the realised profile for the tth type by the addition of η_{vct}. We refer to $\mu + \gamma_{vc}\theta_{vc}+\eta_{vct}$ as the observed profile. This is obscured by additive measurement error ε_{vctr}, which differs for each replicate. The profiles are drawn as lines merely for clarity, because any permutation of variables would leave inference unchanged. One might question the utility of the hidden layer η_{vct}, but our experience is that it is essential for success in applications.

In (2.1) the random variables η_{vct} and ε_{vctr} are experimental and measurement errors, supposed to be independently sampled from Gaussian distributions with zero mean, and variances $\sigma_\eta^2 \geq 0$ and $\sigma^2 > 0$, respectively. For convenience, we consider the Gaussian distribution with variance zero to be a degenerate distribution at its mean. The random variable θ_{vc} does not appear when $\gamma_{vc} = 0$ and appears when $\gamma_{vc} = 1$. We suppose that the γ_{vc} follow independent Bernoulli distributions with success probability p.

Examples of spike-and-slab distributions of the observed profiles obtained by different distributional assumptions for the true effects are shown in Figure 2.4.

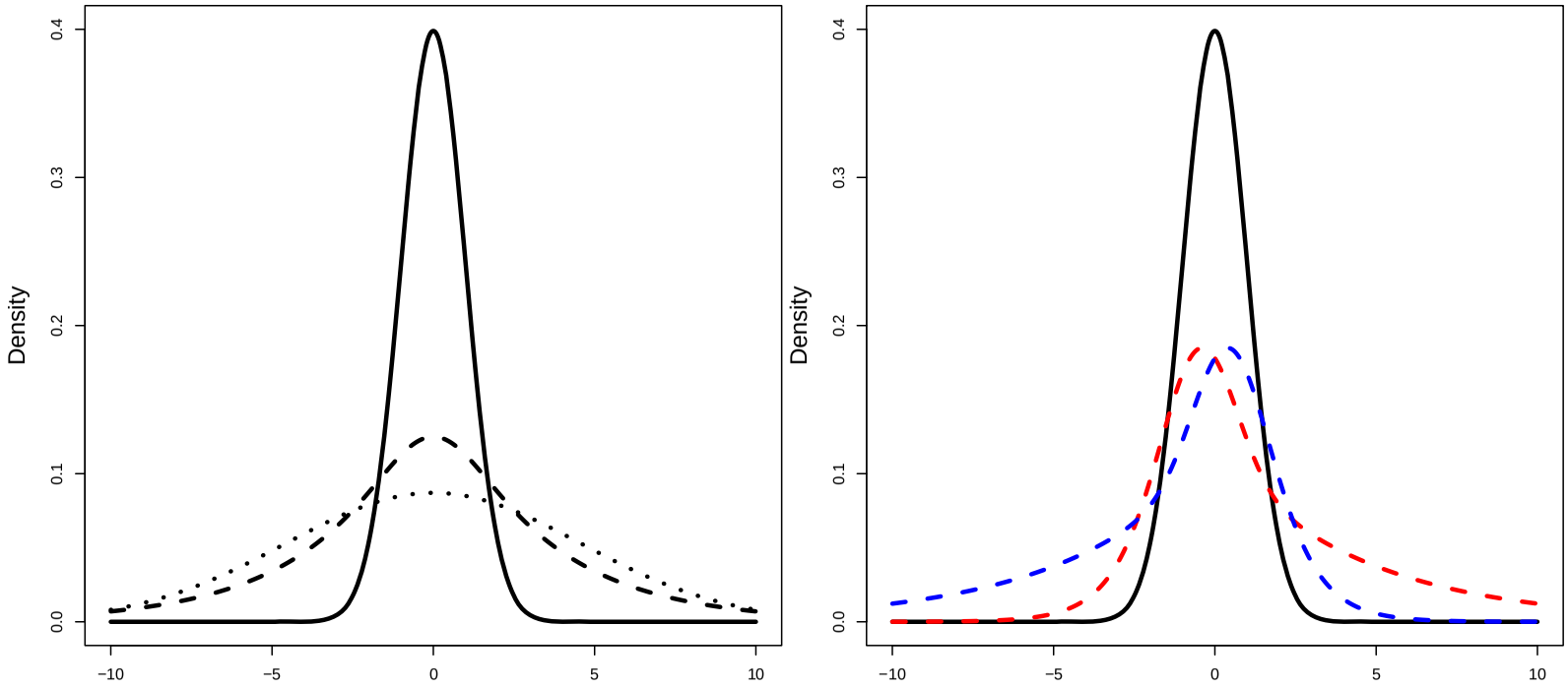

Figure 2.4: Examples of a Gaussian spike (solid) and slab (dotted and dashed) densities; the dotted density is obtained by convolution of two Gaussian distributions; the dashed density is derived by convolution of a Gaussian with a double exponential distribution (left panel). A Gaussian spike versus a right-skewed (red dashed) and a left-skewed (blue dashed) slab densities obtained by convolution of a Gaussian with an asymmetric Laplace distribution (right panel).

It is natural to assume Gaussian distributions for η_{vct} and ε_{vctr}, because they are errors, but we suggest a choice of distributions for the true effects, θ_{vc}. In order to implement a fast classification method we consider models with an analytically closed form joint density for the data y_{vctr} and hence in (2.1) we restrict ourselves to Gaussian and asymmetric Laplace distributions for the true effects.

More details about the estimation of model parameters and applying classification using model (2.1) are presented in Sections 2.2.2 and 2.2.3.

Apart from y_{vctr}, which is supposed to be univariate, the letter y with fewer indices refers to an appropriate vector of data; for example y_v denotes the data of variable v, y_{vc} is the vector of data in class c measured on variable v, and so on. The vector y_c in classical discriminant analysis is supposed to follow a multivariate Gaussian distribution, and vectors y_{ct} are independent realisations of the model assumed for class c.

2.2.2 Gaussian Effects Model

A common model for additive effects is the Gaussian model, so we assume a Gaussian distribution with mean zero and variance $\sigma_\theta^2 > 0$ for the true effects θ_{vc}, in (2.1).

In order to facilitate the calculation of the joint density, we may write model (2.1) in hierarchical form as

$$\begin{aligned} y_{vctr} \mid \eta'_{vct} &\overset{\text{iid}}{\sim} N(\eta'_{vct}, \sigma^2), \\ \eta'_{vct} \mid \theta'_{vc} &\overset{\text{iid}}{\sim} N(\theta'_{vc}, \sigma_\eta^2), \\ \theta'_{vc} \mid \gamma'_{vc} &\overset{\text{iid}}{\sim} N(\mu, \gamma'_{vc}\sigma_\theta^2), \\ \gamma'_{vc} &\overset{\text{iid}}{\sim} B(p), \end{aligned} \tag{2.2}$$

$$\sigma^2, \sigma_\theta^2 > 0, \quad \sigma_\eta^2 \geq 0, \quad 0 < p < 1, \quad \mu \in \mathbb{R},$$

$$v = 1, \ldots, V, \quad c = 1, \ldots, C, \quad t = 1, \ldots, T_c, \quad r = 1, \ldots, R_{ct},$$

where $N(\mu, \sigma^2)$ represents the univariate Gaussian distribution with mean μ and variance σ^2, and $B(p)$ denotes the Bernoulli distribution with success probability p.

In order to estimate the model parameters $\varphi = (\sigma^2, \sigma_\eta^2, \sigma_\theta^2, \mu, p)$ using maximum likelihood, we must calculate the joint density of the data. We do this under a fully marginal model, which provides a universal and automatic set of parameter estimates. If information was available about the indicator variables, better estimates could be obtained.

In many cases each class consists of a single type, and we may drop the index t and write the joint density as

$$f(y;\varphi) = \prod_{v=1}^{V}\prod_{c=1}^{C} f(y_{vc};\varphi), \tag{2.3}$$

where y_{vc} is the data vector of class c measured on variable v,

$$f(y_{vc};\varphi) = pf_1(y_{vc};\varphi) + (1-p)f_0(y_{vc};\varphi), \tag{2.4}$$

$$\begin{aligned} f_0(y_{vc};\varphi) &= (2\pi)^{-R_c/2}\sigma^{1-R_c}(R_c\sigma_\eta^2+\sigma^2)^{-1/2} \\ &\quad \times \exp\left[-\frac{1}{2\sigma^2}\left\{\sum_{r=1}^{R_c} y_{vcr}^2 - R_c\overline{y}_{vc}^2\right\} - \frac{(\overline{y}_{vc}-\mu)^2}{2(\sigma_\eta^2+\sigma^2/R_c)}\right], \end{aligned} \tag{2.5}$$

$$\begin{aligned} f_1(y_{vc};\varphi) &= (2\pi)^{-R_c/2}\sigma^{1-R_c}\{R_c(\sigma_\eta^2+\sigma_\theta^2)+\sigma^2\}^{-1/2} \\ &\quad \times \exp\left[-\frac{1}{2\sigma^2}\left\{\sum_{r=1}^{R_c} y_{vcr}^2 - R_c\overline{y}_{vc}^2\right\} - \frac{(\overline{y}_{vc}-\mu)^2}{2(\sigma_\theta^2+\sigma_\eta^2+\sigma^2/R_c)}\right], \end{aligned} \tag{2.6}$$

in which $\overline{y}_{vc} = R_c^{-1}\sum_{r=1}^{R_c} y_{vcr}$. For details of the calculation, see Section 2.5.6. Given the training data, (2.3) can be maximised to estimate φ.

In order to maximise the log-marginal likelihood using unconstrained optimisation procedures, we implemented a reparametrisation. The log transformation is used for the variance hyper-parameters $\sigma^2, \sigma_\eta^2, \sigma_\theta^2$, the identity for μ, and the logit for p. Asymptotic standard errors for the estimated transformed hyper-parameters maybe computed using the Hessian matrix, given by the optimisation routine, and can be transformed to the original scale using the delta method (Davison, 2003, pp. 33–34). For more precise asymptotic confidence intervals the profile likelihood method can be applied.

Our proposed model (2.2) is unidentifiable in unreplicated situations, that is when $R_c = 1$ for all $c = 1, \ldots, C$. However, $\sigma^2 + \sigma_\eta^2$ is estimable. Setting $\sigma_\eta^2 = 0$, that is ignoring the experimental error layer η_{vct}, we may estimate the remaining parameters.

After fixing the model parameters, calculation of the posterior classification probabilities is straightforward using Bayes' theorem. In order to write the classification posterior formally, we define a discrete random variable $u = 1, \ldots, C$, an auxiliary variable to classify the new observation y^* to one of the already observed classes. Denoting the training set by y, the data in class c by y_c, the data in class c measured on variable v by y_{vc}, and the prior classification probabilities by $\Pr(u = c)$, we can write

$$\begin{aligned}
\Pr(u = c \mid y, y^*) &= k^{-1} \Pr(u = c) f(y, y^* \mid u = c) \\
&= k^{-1} \Pr(u = c) f(y_c, y^*) \prod_{c' \neq c} f(y_{c'}) \\
&= k^{-1} \Pr(u = c) \prod_{v=1}^{V} \left[f(y_{vc}, y_v^*) \prod_{c' \neq c} f(y_{vc'}) \right], \quad (2.7)
\end{aligned}$$

where the last equality holds because the model imposes independent variables, and

$$k = \sum_{c=1}^{C} \Pr(u = c) \prod_{v=1}^{V} \left\{ f(y_{vc}, y_v^*) \prod_{c' \neq c} f(y_{vc'}) \right\} > 0. \quad (2.8)$$

In (2.7) the density $f(y_{vc})$ is the same as in (2.4), and $f(y_{vc}, y_v^*)$ is the joint distribution of the training data in class c measured on variable v with the new data and can be written as

$$f(y_{vc}, y_v^*) = p f_1(y_{vc}, y_v^*) + (1 - p) f_0(y_{vc}) f_0(y_v^*). \quad (2.9)$$

In (2.9), $f_1(y_{vc}, y_v^*)$ is the joint density when the variable-class combination is active, that is both y_{vc} and y_v^* share the same true effect θ_{vc} but may arise with different experimental errors η_{vct} $(t = 1, 2)$. Supposing $\mathbf{1}$ denotes a unit vector having length $R^* + R_c$, then analytical calculations in Section 2.5.4 show that $f_1(y_{vc}, y_v^*)$ corresponds to a multivariate Gaussian density with mean vector $\mu\mathbf{1}$ and $(R^* + R_c) \times (R^* + R_c)$ covariance matrix with diagonal elements $\sigma^2 + \sigma_\eta^2 + \sigma_\theta^2$ and off-diagonal elements $\sigma_\eta^2 + \sigma_\theta^2$ or σ_θ^2. If two observations belong to the same type the off-diagonal element of the covariance matrix equals $\sigma_\eta^2 + \sigma_\theta^2$, and it equals σ_θ^2 otherwise.

2.2.3 Asymmetric Laplace Effects Model

In this section we assume a heavy-tailed and asymmetric distribution for the true effects motivated by the metabolite data. For example in Figure 2.9 we observe extreme peaks in metabolite *maltose.MX1* for *mex1* and *dpe2*, and both peaks are positive. This suggests using a heavy-tailed and asymmetric distribution for the true effects.

The Laplace effects model is similar to the Gaussian effects model, but the asymmetric Laplace distribution allows a more flexible model. An asymmetric Laplace variable can be constructed by flipping a fair coin, if the result is a head we take $-X_L$ and if the result is a tail we take X_R, where X_L and X_R are independent exponentially distributed random variables, with rates $\sigma_{\theta_L}^{-1}$, $\sigma_{\theta_R}^{-1} > 0$ both shifted to $\mu \in \mathbb{R}$. This resulting random variable will have a heavy-tailed distribution with median μ and variance $\sigma_\theta^2 = \sigma_{\theta_L}^2 + \sigma_{\theta_R}^2$; $\log(\sigma_{\theta_R}^2/\sigma_{\theta_L}^2)$ measures the asymmetry. The symmetric Laplace (double exponential) distribution emerges when $\sigma_{\theta_R}^2 = \sigma_{\theta_L}^2$, and the distribution is right-skewed when $\sigma_{\theta_R}^2 > \sigma_{\theta_L}^2$, see Figure 2.5.

Denoting the asymmetric Laplace distribution having median μ, the left tail variance $\sigma_{\theta_L}^2$ and the right tail variance $\sigma_{\theta_R}^2$ by $L(\mu, \sigma_{\theta_L}^2, \sigma_{\theta_R}^2)$, we may write the asymmetric Laplace model in hierarchical form similar to (2.2) except that the distribution of θ'_{vc} given γ'_{vc} is replaced by

$$\theta'_{vc} \mid \gamma'_{vc} \overset{\text{iid}}{\sim} L(\mu, \gamma'_{vc}\sigma_{\theta_L}^2, \gamma'_{vc}\sigma_{\theta_R}^2), \quad \sigma_{\theta_L}^2, \sigma_{\theta_R}^2 > 0, \quad \mu \in \mathbb{R},$$

and for convenience we consider $L(\mu, 0, 0)$ to be a distribution degenerate at μ.

The model parameters can be obtained by maximising the likelihood func-

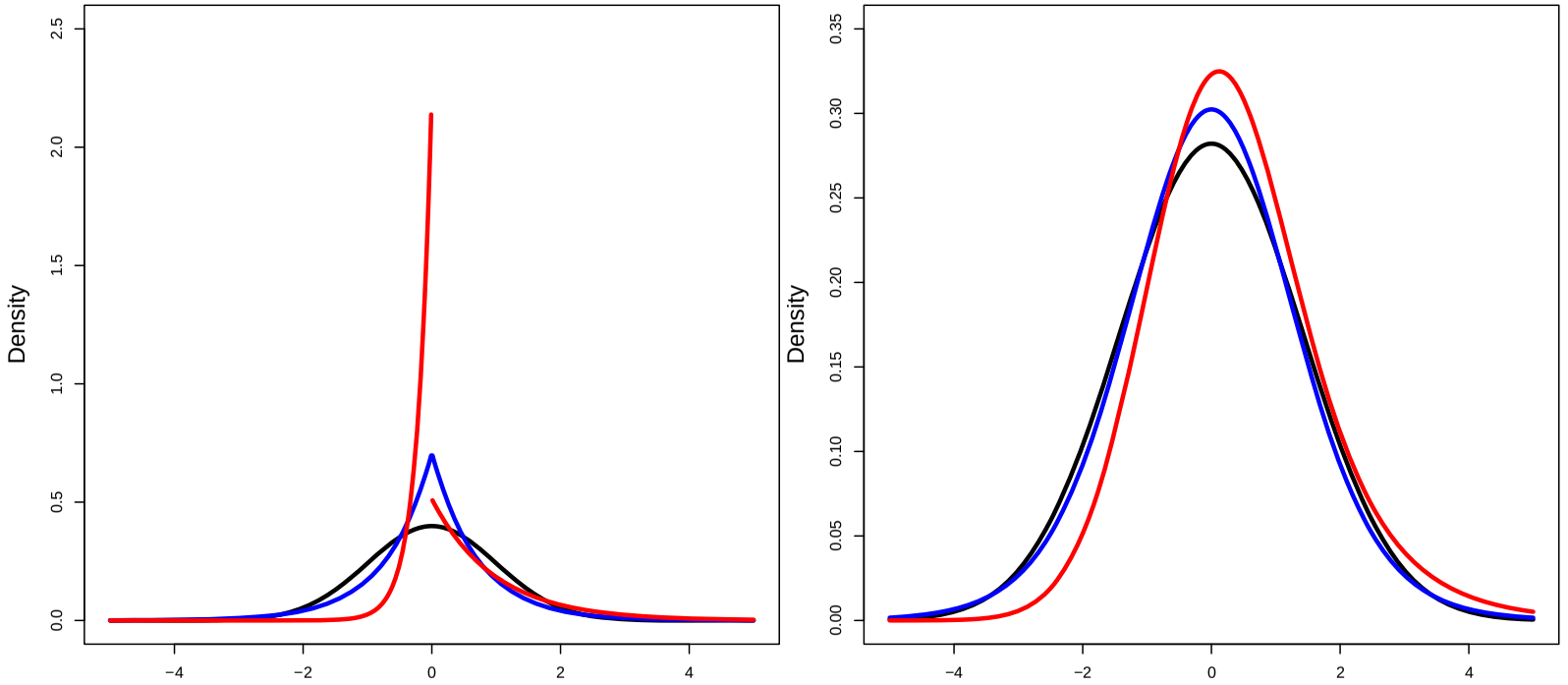

Figure 2.5: The standard Gaussian density (black), the symmetric Laplace density (blue), and the asymmetric Laplace density with $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 10$ (red), all having zero median and unit variance, giving examples of the true effect distribution, (left panel). The density obtained by convolution of the left panel densities with a standard Gaussian density, giving examples of the observed profile distribution when the variable-class combination is active (right panel).

tion (2.3), but in (2.4) $f_1(y_{vc})$ is replaced by

$$\begin{aligned} f_1(y_{vc}) &= 2^{-1}(2\pi\sigma^2)^{-(R_c-1)/2} {R_c}^{-1/2} \\ &\quad \times \exp\left(-\frac{1}{2\sigma^2}\sum_{r=1}^{R_c} y_{vcr}^2\right)(I_L + I_R), \qquad (2.10) \\ I_L &= \sigma_{\theta_L}^{-1} \exp\left\{\frac{R_c}{2\sigma^2}\left(\overline{y}_{vc} + \frac{\sigma^2}{R_c\sigma_{\theta_L}}\right)^2 + \frac{\sigma_\eta^2}{2\sigma_{\theta_L}^2} - \frac{\mu}{\sigma_{\theta_L}}\right\} \\ &\quad \times \Phi\left(\frac{\mu - \overline{y}_{vc} - \sigma^2/(R_c\sigma_{\theta_L}) - \sigma_\eta^2/\sigma_{\theta_L}}{\sqrt{\sigma_\eta^2 + \sigma^2/R_c}}\right), \\ I_R &= \sigma_{\theta_R}^{-1} \exp\left\{\frac{R_c}{2\sigma^2}\left(\overline{y}_{vc} - \frac{\sigma^2}{R_c\sigma_{\theta_R}}\right)^2 + \frac{\sigma_\eta^2}{2\sigma_{\theta_R}^2} + \frac{\mu}{\sigma_{\theta_R}}\right\} \\ &\quad \times \Phi\left(\frac{\overline{y}_{vc} - \mu - \sigma^2/(R_c\sigma_{\theta_R}) - \sigma_\eta^2/\sigma_{\theta_R}}{\sqrt{\sigma_\eta^2 + \sigma^2/R_c}}\right), \end{aligned}$$

where Φ denotes the standard Gaussian cumulative distribution function; for details see Section 2.5.6.

The posterior classification probabilities for the asymmetric Laplace model can be obtained using Bayes' theorem and are similar to (2.7) and (2.9) but with

$$\begin{aligned} f_1(y_{vc}, y_v^*) &= k_0(k_L I_L + k_R I_R), \qquad (2.11) \\ k_0 &= (2\pi)^{-(R_c+R^*)/2}\sigma_\eta^{-1}\pi^{1/2} \exp\left\{-\frac{1}{2\sigma^2}\left(\sum_{r=1}^{R_c} y_{vcr}^2 + \sum_{r=1}^{R^*} {y_{vr}^*}^2\right)\right\} |\mathbf{A}|^{-1/2}, \\ k_L &= (2\sigma_{\theta_L})^{-1} \exp\left\{\sigma_\eta^2/(4\sigma_{\theta_L}^2) - \mu/\sigma_{\theta_L}\right\}, \\ I_L &= \exp\left(\frac{1}{2}\mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L\right)\Phi\left(\frac{c_L + \mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L}{\sqrt{1 + \mathbf{d}_L\mathbf{A}^{-1}\mathbf{d}_L}}\right), \\ k_R &= (2\sigma_{\theta_R})^{-1} \exp\left\{\sigma_\eta^2/(4\sigma_{\theta_R}^2) - \mu/\sigma_{\theta_R}\right\}, \\ I_R &= \exp\left(\frac{1}{2}\mathbf{b}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R\right)\Phi\left(\frac{c_R + \mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R}{\sqrt{1 + \mathbf{d}_R\mathbf{A}^{-1}\mathbf{d}_R}}\right), \end{aligned}$$

where $\mathbf{A}$ is a 2×2 symmetric positive-definite matrix,

$$\mathbf{A} = \begin{bmatrix} R_c/\sigma^2 + 1/(2\sigma_\eta^2) & -1/(2\sigma_\eta^2) \\ -1/(2\sigma_\eta^2) & R^*/\sigma^2 + 1/(2\sigma_\eta^2) \end{bmatrix},$$

$|\mathbf{A}|$ is the determinant of $\mathbf{A}$, $\mathbf{b}_L, \mathbf{b}_R, \mathbf{d}_L, \mathbf{d}_R$ are 2×1 vectors, and c_L and c_R are constants. Denoting the number of replicates of the new observed data with R^* and $\overline{y}_v^* = R^{*-1} \sum_{r=1}^{R^*} y_{vr}^*$, we may write

$$\begin{aligned}
\mathbf{b}_L &= [R_c \overline{y}_{vc}/\sigma^2 + 1/(2\sigma_{\theta_L}) \ , \ R^* \overline{y}_v^*/\sigma^2 + 1/(2\sigma_{\theta_L})]^{\mathrm{T}}, \\
\mathbf{b}_R &= [R_c \overline{y}_{vc}/\sigma^2 - 1/(2\sigma_{\theta_R}) \ , \ R^* \overline{y}_v^*/\sigma^2 - 1/(2\sigma_{\theta_R})]^{\mathrm{T}}, \\
c_L &= \frac{\mu - \sigma_\eta^2/(2\sigma_{\theta_L})}{\sqrt{\sigma_\eta^2/2}}, \ c_R = -\frac{\mu + \sigma_\eta^2/(2\sigma_{\theta_R})}{\sqrt{\sigma_\eta^2/2}}, \\
\mathbf{d}_L &= \left[-1/\sqrt{2\sigma_\eta^2} \ , \ -1/\sqrt{2\sigma_\eta^2}\right]^{\mathrm{T}}, \ \mathbf{d}_R = \left[1/\sqrt{2\sigma_\eta^2} \ , \ 1/\sqrt{2\sigma_\eta^2}\right]^{\mathrm{T}}.
\end{aligned}$$

For calculation details for $f_1(y_{vc}, y_v^*)$, see Section 2.5.4.

2.3 Extensions

2.3.1 Unobserved Class Probability

In classification it is often assumed that the new observation belongs to one of the classes already observed. However in many applications it is meaningful to allow the new subject to belong to a new group; in machine learning such statistical procedures are called semi-supervised learning. In the Bayesian paradigm calculation of the posterior probability that the new observation make a new group is straightforward using Bayes' theorem and can be implemented easily for any model-based classifier. We may allow the discrete random variable u in (2.7) to take value $C + 1$ also, thus denoting that the new observation belongs to a new group. Therefore we just need to consider $\Pr(u = C + 1 \mid y, y^*)$ in (2.7) and recalculate the normalising

constant k in (2.8):

$$\begin{aligned}\Pr(u = C+1 \mid y, y^*) &= k^{-1} \Pr(u = C+1) f(y, y^* \mid u = C+1) \\ &= k^{-1} \Pr(u = C+1) f(y^*) \prod_{c=1}^{C} f(y_c) \\ &= k^{-1} \Pr(u = C+1) \prod_{v=1}^{V} \left[f(y_v^*) \prod_{c=1}^{C} f(y_{vc}) \right],\end{aligned}$$

in which $f(y_{vc})$ is the joint distribution of the data in class c and variable v calculated in (2.4). The density $f(y_v^*)$ can be calculated in the same way, by replacing $\sum_{r=1}^{R_c} y_{vcr}^2$ with $\sum_{r=1}^{R^*} y_v^{*2}$ and $\overline{y}_{vc}$ with $R^{*-1} \sum_{r=1}^{R^*} y_v^*$. It is required to assume a prior classification probability for y^* being a new class, $\Pr(u = C+1)$. The normalising constant that makes the posterior classification probabilities sum to one now also includes the probability for a new type. Setting $f(y_{C+1}, y^*) = f(y_{C+1}) f(y^*)$ and $f(y_{vC+1}, y_v^*) = f(y_v^*) f(y_{vC+1})$ we may write

$$\Pr(u = c \mid y, y^*) = \frac{\Pr(u = c) \left[f(y_c, y^*) \prod_{c' \neq c} f(y_{c'}) \right]}{\sum_{i=1}^{C+1} \Pr(u = i) \left[f(y_i, y^*) \prod_{c' \neq i} f(y_{c'}) \right]} \tag{2.12}$$

$$= \frac{\Pr(u = c) \prod_{v=1}^{V} \left[f(y_{vc}, y_v^*) \prod_{c' \neq c} f(y_{vc'}) \right]}{\sum_{i=1}^{C+1} \Pr(u = i) \prod_{v=1}^{V} \left[f(y_{vi}, y_v^*) \prod_{c' \neq i} f(y_{vc'}) \right]}. \tag{2.13}$$

2.3.2 Built-in Variable Selection

Spike-and-slab models are considered as a general tool for Bayesian variable selection in regression, by considering a mixture of a point mass at zero (the spike) and a continuous distribution with tails away from zero (the slab) for regression parameters. The problem of Bayesian variable selection is mainly computational, because it requires implementation of trans-dimensional Markov chain Monte Carlo methods, which is computationally challenging and slow. The Bayesian variable selection procedure developed by George and McCulloch (1997) assumes a mixture of two Gaussian distributions, giving a mixture of distributions having the same support, and

so allowing a simple Gibbs sampler to sample from the posterior distribution. In the Bayesian variable selection of George and McCulloch (1997), the mixture prior (the Gaussian spike-and-slab model) may be constructed by convolution of a mass point with Gaussian noise (the spike distribution) and convolution of another distribution (the effects distribution) with Gaussian noise (the slab distribution). This insight generalises the classical Bayesian variable selection of George and McCulloch (1997) that uses a Gaussian distribution for both spike and slab distributions. Our approach gives always a Gaussian spike prior but produces a variety of slab distributions depending on the distribution assumed for effects. If both noise and effect distributions are Gaussian the mixture prior is that of George and McCulloch (1997), see Figures 2.4 and 2.5.

Integrating over the spike and slab prior in our proposed models solves the curse of dimensionality in classification, because negligible effects are likely to come from the spike prior which does not affect the classification, while large effects are more likely to be generated from the slab prior, which guides the classification. This produces classification procedures which are less sensitive to uninformative variables, because when a variable is useless, $f_1(y_{vc}, y_v^*)$, for all $c = 1, \ldots, C$, is much smaller than $f_0(y_{vc})f_0(y_v^*)$, hence $f(y_{vc}, y_v^*)$ in (2.9) is close to $f_0(y_{vc})f_0(y_v^*)$, and consequently the posterior classification probabilities in (2.7) shrink toward the prior.

Our proposed spike-and-slab models in Sections 2.2.2 and 2.2.3 select variable-class combinations instead of variables. We may implement variable selection in addition to variable-class selection by introducing another Bernoulli variable δ_v to control appearance of the variable-class effect,

$$y_{vctr} = \mu + \eta_{vct} + \delta_v \gamma_{vc} \theta_{vc} + \varepsilon_{vctr}. \tag{2.14}$$

The hierarchical version of (2.14) for the Gaussian effects model may be

written as

$$\begin{aligned} y_{vctr} \mid \eta'_{vct} &\overset{\text{iid}}{\sim} N(\eta'_{vct}, \sigma^2), \\ \eta'_{vct} \mid \theta'_{vc} &\overset{\text{iid}}{\sim} N(\theta'_{vc}, \sigma^2_\eta), \\ \theta'_{vc} \mid \gamma'_{vc} &\overset{\text{iid}}{\sim} N(\mu, \gamma'_{vc}\sigma^2_\theta), \\ \gamma'_{vc} \mid \delta'_v &\overset{\text{iid}}{\sim} B(\delta'_v p), \\ \delta'_v &\overset{\text{iid}}{\sim} B(q), \end{aligned} \tag{2.15}$$

$$\sigma^2, \sigma^2_\theta > 0 \quad \sigma^2_\eta \geq 0, \quad 0 < p < 1, \quad 0 < q \leq 1, \quad \mu \in \mathbb{R},$$

$$v = 1, \ldots, V, \quad c = 1, \ldots, C, \quad t = 1, \ldots, T_c, \quad r = 1, \ldots, R_{ct},$$

Model (2.14) is a generalisation of (2.2); they are identical if $\delta_v = 1$ with probability 1 for all $v = 1, \ldots, V$, or equivalently if $q = 1$. If variable v is active ($\delta_v = 1$) variable-class selection is allowed, and if variable v is inactive ($\delta_v = 0$) the model imposes a degenerate distribution for the true effects θ_{vc} for all $c = 1, \ldots, C$. Hence, q is the proportion of active variables, and p is the proportion of active variable-class combinations for active variables. This gives pq active variable-class combinations in total. For a graphical representation of the variable selection model, see Figure 2.6.

The hierarchical representation of model (2.14) for the asymmetric Laplace model is straightforward, because the only difference between the Gaussian model and the asymmetric Laplace model is the effect distribution. Hence, in order to obtain the asymmetric Laplace model with variable selection, in (2.15), we replace $N(\mu, \gamma'_{vc}\sigma^2_\theta)$ with $L(\mu, \gamma'_{vc}\sigma^2_{\theta_L}, \gamma'_{vc}\sigma^2_{\theta_R})$, $\sigma^2_{\theta_R}, \sigma^2_{\theta_L} > 0$.

The variable selection generalisation of the Gaussian and the asymmetric Laplace models still have analytically closed form marginal posteriors, of form

$$f(y; \varphi) = \prod_{v=1}^{V} \left\{ q \left[\prod_{c=1}^{C} p f_1(y_{vc}) + (1-p) f_0(y_{vc}) \right] + (1-q) \prod_{c=1}^{C} f_0(y_{vc}) \right\}, \tag{2.16}$$

where $\varphi = (\sigma^2, \sigma^2_\eta, \sigma^2_\theta, \mu, p, q)$ for the Gaussian model and $\varphi = (\sigma^2, \sigma^2_\eta, \sigma^2_{\theta_L}, \sigma^2_{\theta_R}, \mu, p, q)$ for the asymmetric Laplace model. The densities $f_0(y_{vc})$ and $f_1(y_{vc})$ are defined in (2.6) for the Gaussian and in (2.10) for the asymmetric Laplace model.

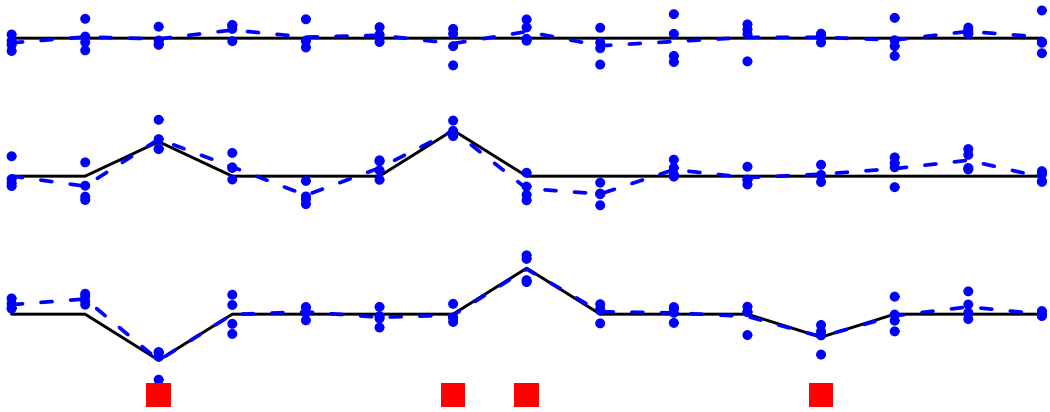

Figure 2.6: The variable selection model (2.14) represented on 15 variables with true profile (solid black), observed profile (dashed), and four replications for each variable-class combination (solid blobs). For the active variables, a red square is plotted.

The posterior classification probability can be calculated from (2.13) by replacing $f(y_{vc}, y_v^*)$ with

$$f(y_{vc}, y_v^*) = q\,[p f_1(y_{vc}, y_v^*) + (1-p) f_0(y_{vc}) f_0(y_v^*)] + (1-q) f_0(y_{vc}) f_0(y_v^*), \tag{2.17}$$

in which the density $f_1(y_{vc}, y_v^*)$, is defined in (2.9) for the Gaussian and in (2.11) for the asymmetric Laplace model. The posterior classification probability (2.17) is a convex combination of two densities: a density that guides the classification, f_1, and another that is a random classifier, f_0. The convex combination appears in two levels: in the variable level, controlled by q, and in the variable-class level, controlled by p. As an immediate consequence, when p equals zero, that is, all variable-class combination are inactive, the posterior classification probabilities equal those for the prior. For $q = 1$ the variable selection model reduces to the Gaussian and the asymmetric Laplace model explained in Sections 2.2.2 and 2.2.3. Setting $q = 0$ allows no active variable-class combinations for any variable, so the posterior classification probabilities equal those for the prior. According to our experience, greater values of p and q often yield more certain posterior classification probabilities. However, when σ_θ^2 or $\sigma_{\theta_L}^2 + \sigma_{\theta_R}^2$ is chosen or estimated to be considerably smaller than σ_η^2, the posterior probabilities shrink toward the prior.

In Bayesian hypothesis testing, the posterior odds for alternative hypothesis relative to null hypothesis depends on data only through the Bayes factor, say B^{10}. Assuming a prior probability 1/2 that the alternative hypothesis is true, the posterior probability that the alternative hypothesis is true will be $B^{10}/(1+B^{10})$. Kass and Raftery (1995) propose the following scale for the Bayes factor B^{10} as an evidence against the null hypothesis: negative if $\log B^{10} \leq 0$, hardly worth a mention if $0 < \log B^{10} \leq 1$, positive if $1 < \log B^{10} \leq 3$, strong if $3 < \log B^{10} \leq 5$, and very strong if $\log B^{10} > 5$. Hence, the main advantage of the variable selection model is giving an importance measure for variables, B_v^{10}, and for variable-class combinations, B_{vc}^{10},

$$\begin{aligned} \log B_v^{10} &= \log f(y_v \mid \delta_v = 1) - \log f(y_v \mid \delta_v = 0), \\ \log B_{vc}^{10} &= \log f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) - \log f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 0). \end{aligned} \tag{2.18}$$

The Bayes factor can be regarded as a weight that sorts variables and variable-class combinations, showing which ones guide the classification more than others. Though the scale proposed by Kass and Raftery (1995) is arbitrary, but is broadly accepted and often used in applications.

The quality of estimation of φ depends on the data structure. Often when $\sigma_\theta^2/\sigma_\eta^2$ is small, it is hard to estimate p and q precisely. A small number of variables V also yields estimates of q with large uncertainty. A small number of variable-class combinations VC affects efficiency of estimation of the hyper-parameter p. When the training data are unreplicated, that is $R_c = 1$ for $c = 1 \ldots C$, both Gaussian and asymmetric Laplace models are unidentifiable, but $\sigma^2 + \sigma_\eta^2$ is identifiable. For more details see Section 4.4.

2.4 Examples

2.4.1 Metabolite Data

First we implement our proposed classification methods on the metabolite data of Section 1.3.1 to answer questions such as

- The metabolite behaviour of the unknown types *d172, d263, sex3,* and *ke103* is closest to which of the known types?
- How similar are these unknown plants to the known types?
- Is it possible that mutants come from a category that is not in the known types provided in the training set?
- Which metabolites can characterise known and unknown types?
- Which known and unknown mutants follow similar metabolite patterns?

We answer the questions using the methodology provided in this chapter, except for the last question which requires clustering and will be discussed in Chapter 3. We use the Bayes factor B_v^{10} to assess which metabolites are useful for classification. Metabolites that characterise mutants must be informative for classification, so metabolites that are useless for classification do not characterise the types.

Before applying classification procedures, the vector φ must be estimated. First, we consider taking $q = 1$, as described in Sections 2.2.2 and 2.2.3. The model parameters are estimated by maximising the likelihood (2.3). The negative profile likelihoods, which are useful to obtain confidence intervals for the estimated parameters, are reported in Figure 2.7. The estimated parameters with their standard errors in parentheses, derived by the delta method, are shown in Table 2.1 for the Gaussian and the asymmetric Laplace models. Table 2.1 shows that the general mean estimate $\hat{\mu}$ is about the same using the Gaussian and the asymmetric Laplace models. The experimental error layer must be included because zero does not lie in the 95% confidence interval for $\hat{\sigma}_\eta^2$. The estimated measurement error variances are the same using the Gaussian and the asymmetric Laplace models. The estimated proportion of active variable-class combinations for the asymmetric Laplace model p is larger than for the Gaussian model, but the two 95% confidence intervals overlap. The 95% confidence intervals for the hyper-parameter p using the profile likelihoods are $(0.015, 0.064)$ and $(0.038, 0.134)$ for the Gaussian and the asymmetric Laplace models, respectively. The profile likelihood confidence interval for p is wider for the asymmetric Laplace model, suggesting

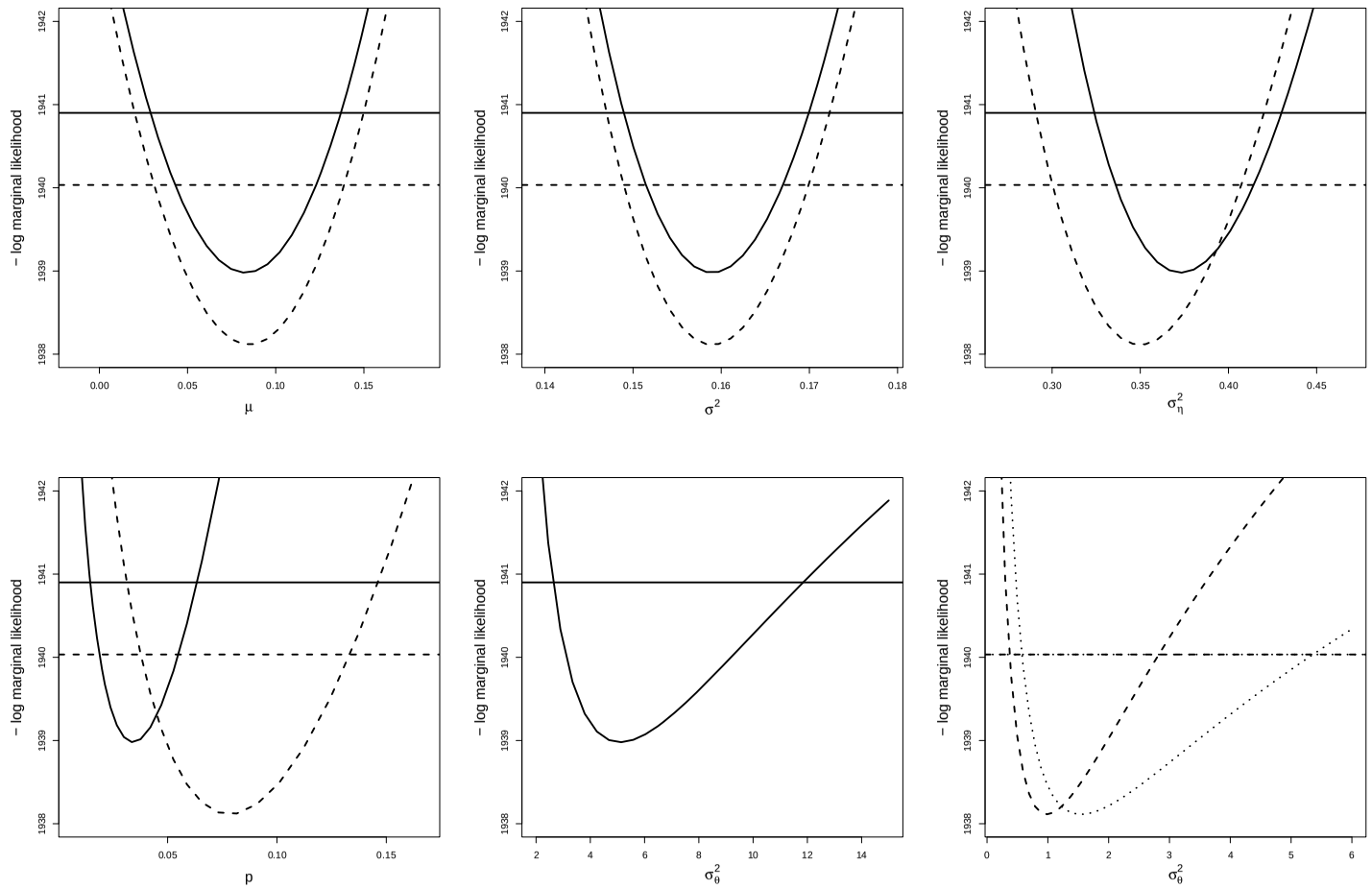

Figure 2.7: Negative profile likelihoods for the parameters using the Gaussian model (solid) and the asymmetric Laplace model (dashed). The 95% confidence intervals can be found by cutting the profile likelihood curves with the corresponding horizontal lines. The bottom middle panel corresponds to the variance of the Gaussian model. The bottom right panel corresponds to the parameters $\sigma^2_{\theta_L}$ (dashed line) and $\sigma^2_{\theta_R}$ (dotted line) of the asymmetric Laplace model.

	$\hat{\sigma^2}$	$\hat{\sigma}^2_\eta$	$\hat{\sigma}^2_\theta$		$\hat{\mu}$	$\hat{p}$
			$\hat{\sigma}^2_{\theta_L}$	$\hat{\sigma}^2_{\theta_R}$		
Gaussian	(0.005) 0.159	(0.033) 0.373	(2.773) 5.155		(0.028) 0.083	(0.019) 0.034
Laplace	(0.005) 0.159	(0.043) 0.35	(0.778) 0.983	(2.361) 1.547	(0.028) 0.085	(0.071) 0.078

Table 2.1: Estimated parameters and their respective standard errors, derived by the delta method (above each estimate), for the Gaussian and the asymmetric Laplace models.

that estimation of p using the asymmetric Laplace model is more difficult; a similar pattern is observed in our simulation study reported in Section 4.4. Profile likelihood confidence intervals for other parameters can be obtained by cutting the profile likelihood curves of Figure 2.7 with the reference horizontal lines derived from the chi-square distribution, but they are not very different from confidence intervals obtained using the delta method. In all profile likelihood plots the minimised value of the negative log likelihood for the asymmetric Laplace model is smaller than for the Gaussian model, suggesting that the asymmetric Laplace model fits better. However, the asymmetric Laplace model contains 6 parameters but the Gaussian model involves 5, and a reasonable comparison criterion, should consider the model dimension too. We use the BIC of Schwarz (1978). The BIC for the asymmetric Laplace model equals 3892.06 and for the Gaussian model equals 3891.15, suggesting that the Gaussian model is better.

The posterior classification probabilities are calculated using (2.13) and are reported for the Gaussian model and the asymmetric Laplace model in Table 2.2.

The classification results using the Gaussian and the asymmetric Laplace model are similar. The maxima of the a posteriori classification probabilities appear at the same place. The classification probabilities for the mutant *ColWT* are spread out between the wild types *WsWT, RLDWT* and *tpt*, with a tiny probability of it being a new type. Having almost equal probabilities for *ColWT* to be assigned to the wild types is expected, because *ColWT* belongs to the wild types. Mutant *sex3* almost equally likely to be attributed to the wild types and to *tpt*. The mutants *isa2*, *sex3* and *ColWT* may perhaps

		Gaussian Model							
		WsWT	*RLDWT*	*tpt*	*pgm*	*sex4*	*mex1*	*dpe2*	New
Known	*ColWT*	26.48	24.85	23.19	0.59	13.62	0	0	11.27
	isa2	0.03	0.03	0.02	0.64	82.06	0	0	17.23
	sex1	0.32	0.13	0.73	98.33	0.33	0	0	0.15
Unknown	*d172*	0.4	0.76	0.32	0.58	97.71	0	0	0.23
	d263	4.8	9.2	3.91	7.13	72.62	0	0	2.34
	ke103	6.25	8.21	17.07	56.96	2.05	0	0	9.47
	sex3	24.34	23.19	23.79	0.33	12.95	0	0	15.42
		Asymmetric Laplace Model							
		WsWT	*RLDWT*	*tpt*	*pgm*	*sex4*	*mex1*	*dpe2*	New
Known	*ColWT*	30.61	26.65	23.12	0.56	9.67	0	0	9.39
	isa2	0.01	0.01	0	1.36	92.37	0	0	6.25
	sex1	0.14	0.03	0.24	99.39	0.16	0	0	0.03
Unknown	*d172*	0.03	0.07	0.02	0.1	99.77	0	0	0.01
	d263	1.25	3.08	0.79	4.35	90.11	0	0	0.43
	ke103	8.12	8.47	34.56	41.24	0.91	0	0	6.71
	sex3	27.25	23.84	24.8	0.23	6.95	0	0	16.93

Table 2.2: Posterior classification percentages assuming a uniform prior for the Gaussian model and the asymmetric Laplace model when $q = 1$. The model parameters are estimated from the data, and the maximum a posteriori percentages are shown in red.

be a new unobserved types; *isa2* is relatively certain to be *sex4*; *sex1*, *d172* and *d263* are allocated quite strictly; and *ke103* is identified to be *pgm*, but is also close to *tpt*.

Looking at the metabolite data in Figure 2.9 confirms the classification results of Table 2.2. The wild types, *tpt*, *ke103*, and *sex3* have almost flat profiles. Mutant *d172* has a profile similar to *sex4*. Unknown type *sex1* is similar to known type *pgm*. Known types *dpe2* and *mex1*, have a distinguishable profile on *maltose.MX1*, so it is not surprising to see that all types have very small probabilities to be assigned to *dpe2* and *mex1*.

Classification using the variable selection approach introduced in Section 2.3.2 needs estimation or tuning of the parameter q as well. The estimation of all model parameters together for the variable selection models

Gaussian Variable Selection Model									
		WsWT	*RLDWT*	*tpt*	*pgm*	*sex4*	*mex1*	*dpe2*	New
Known	*ColWT*	30.78	27.84	20.65	0.34	11.17	0	0	9.21
	isa2	0.08	0.08	0.03	1.33	77.76	0	0	20.72
	sex1	2.49	0.44	3.7	90.09	2.85	0	0	0.43
Unknown	*d172*	1.37	3.39	0.88	1.76	92.27	0	0	0.33
	d263	3.01	7.72	1.98	4.44	81.92	0	0	0.92
	ke103	18.59	4.2	41.49	27.31	3.88	0	0	4.52
	sex3	19.59	40.4	13.23	0.06	11.81	0	0	14.92
Asymmetric Laplace Variable Selection Model									
		WsWT	*RLDWT*	*tpt*	*pgm*	*sex4*	*mex1*	*dpe2*	New
Known	*ColWT*	34.04	29.35	20.69	0.44	6.57	0	0	8.92
	isa2	0.28	0.14	0.02	5.36	63.72	0	0	30.48
	sex1	2.53	0.53	1.97	93.04	1.72	0	0	0.21
Unknown	*d172*	0.44	1.06	0.17	0.88	97.33	0	0	0.13
	d263	1.56	3.88	0.73	5.12	87.88	0	0	0.83
	ke103	25.41	6	44.86	20.31	1.57	0	0	1.84
	sex3	16.24	64.7	7.96	0.02	2.08	0	0	9

Table 2.3: Posterior classification percentages for the Gaussian and the asymmetric Laplace variable selection models, assuming a uniform prior. The maximum a posteriori percentages are in red.

is difficult and yields estimates with large uncertainties. We fix parameters $\sigma^2, \sigma_\eta^2, \sigma_\theta^2, \mu$ to the values already estimated by setting $q = 1$, and then estimate p and q. This gives $\hat{p} = 0.458$ and $\hat{q} = 0.156$ for the Gaussian model and $\hat{p} = 0.83$ and $\hat{q} = 0.183$ for the asymmetric Laplace model. Classifications using the variable selection models are reported in Table 2.3. The allocation using the variable selection models agrees with that of Table 2.2, except that the mutant *sex3* in the variable selection models is attributed to *RLDWT*, but in Table 2.2 is identified to be the wild type *WsWT*. Also the mutant *ke103* in Table 2.2 is allocated to *pgm* but in Table 2.3 is classified to *tpt*. This is not surprising because looking at the data in Figure 2.9 we see that the wild types and *tpt* have similar profiles.

		WsWT	*RLDWT*	*tpt*	*pgm*	*sex4*	*mex1*	*dpe2*
Naive Bayes	*ColWT 1*	100	0	0	0	0	0	0
	ColWT 2	0	100	0	0	0	0	0
	ColWT 3	0	100	0	0	0	0	0
Linear	*ColWT 1*	95.6	4.4	0	0	0	0	0
discriminant	*ColWT 2*	0	100	0	0	0	0	0
	ColWT 3	0	100	0	0	0	0	0
Gaussian	*ColWT 1*	16.2	16.2	15.8	12.6	14.2	12.5	12.5
	ColWT 2	16.0	15.8	15.5	12.6	15.1	12.5	12.5
	ColWT 3	16.2	16.3	15.8	12.6	14.2	12.5	12.5
Gaussian	*ColWT 1*	16.8	16.4	15.3	12.5	14.0	12.4	12.4
variable	*ColWT 2*	16.6	16.5	14.9	12.5	14.6	12.5	12.5
selection	*ColWT 3*	16.7	16.7	15.1	12.5	14.1	12.4	12.4

Table 2.4: Classification percentages using the naive Bayes, the linear discriminant, the Gaussian and the Gaussian variable selection procedures assuming a uniform prior for observations of $ColWT$.

According to the Gaussian variable selection model, the metabolites *Maltose.MX1*, *raffinose2*, *X18*, *L.ascorbic*, and *glumatic.3*, have large Bayes factors B_v^{10}, *X16* is relatively important, *serine.3* and *saccharic* have negligible Bayes factors, the rest have negative Bayes factors and hence are unimportant, see Figure 2.8. The Bayes factors B_v^{10} are coded using a heat bar and B_{vc}^{10} are coded by heat blobs in Figure 2.9, confirming that highly important variable-class combinations (red blobs), appear for highly important variables (red bar). The estimated value of q, $\hat{q} = 0.156$, states that about 16% of variables are active, that is about 6 variables out of 43. This agrees with the heat bar: 6 variables have Bayes factors that are red or orange.

Procedures like the linear discriminant or naive Bayes do not usually allow for the possibility of replicated data. This could be dealt with by some form of averaging at the risk of a loss of information. Thus it is not straightforward to get posterior probabilities like Table 2.2 and 2.3. The naive Bayes and the linear discriminant posterior classification probabilities for observations of the mutant $ColWT$ are given in Table 2.4. Even the naive Bayes method gives

over-confident classification probabilities for observations of the wild type $ColWT$, but our approach gives more widely spread probabilities, especially across the other wild types, namely $WsWT$ and $RLDWT$.

The common classifiers are inefficient in high dimensions because of over-fitting (Hand, 2006). The naive Bayes classifier, which is the quadratic discriminant assuming independent variables, often performs better, see Bickel and Levina (2004) for theoretical discussion.

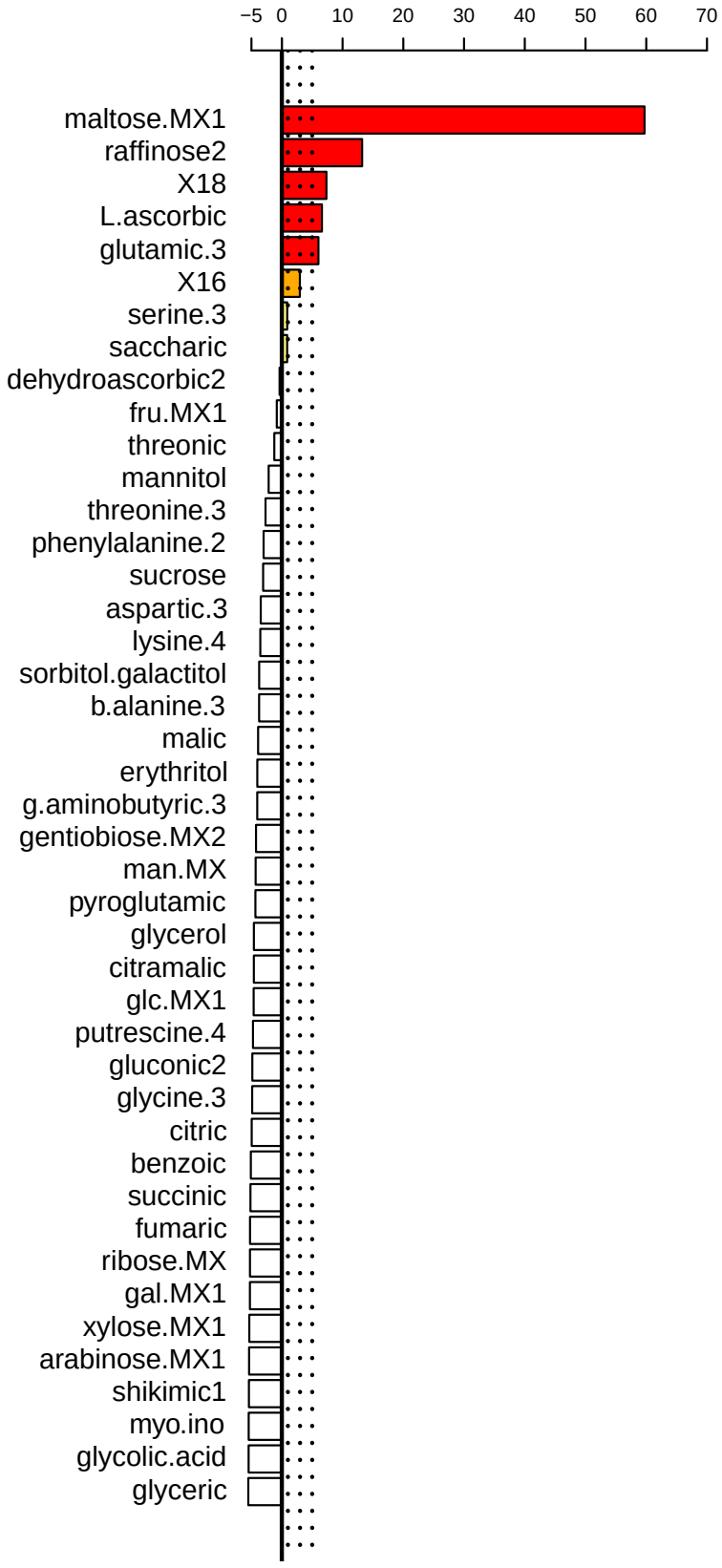

Figure 2.8: The log Bayes factor, $\log B_v^{10}$, for the metabolite data using the Gaussian variable selection model. The horizontal dotted lines represent the values that are used to categorise and colour the log Bayes factors. See also the caption to Figure 2.9.

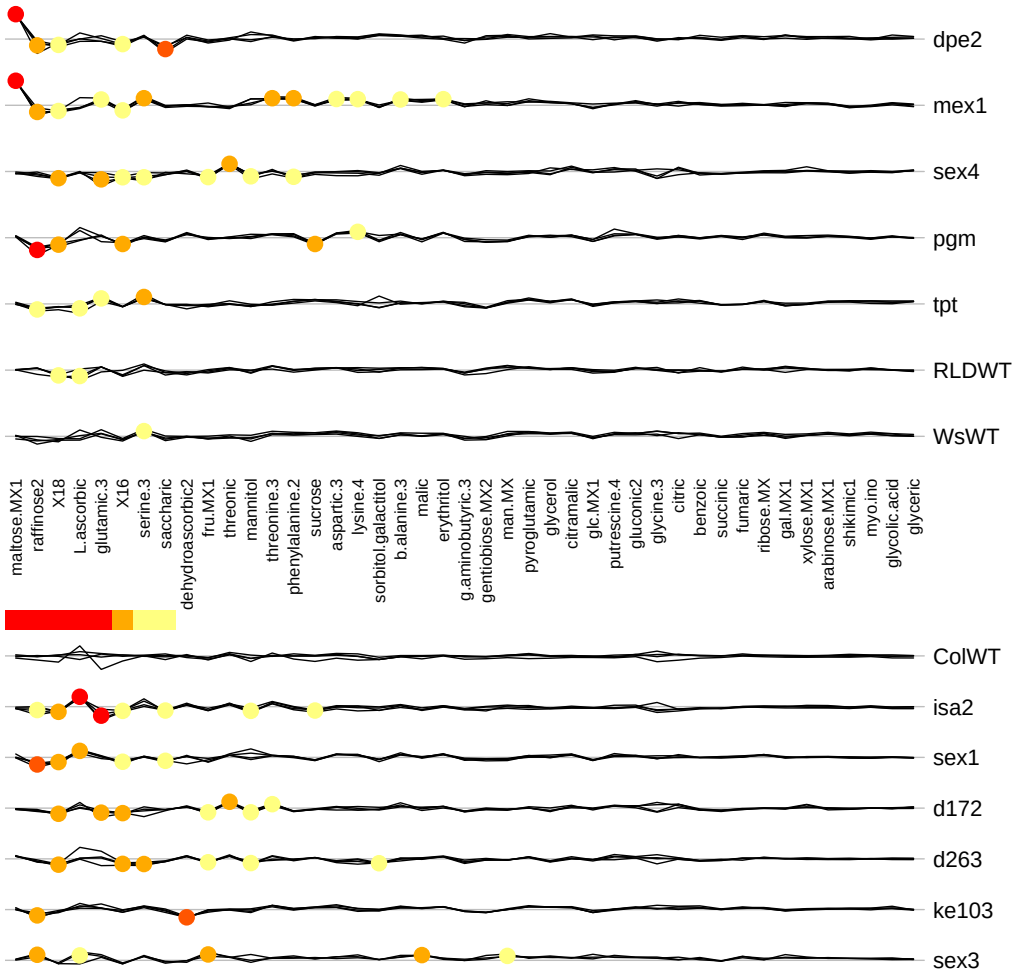

Figure 2.9: Profile plots for the metabolite data represented with variable and variable-class importance obtained using Bayes factors, B^{10}, for the Gaussian effects model. Very important corresponds to red ($\log B^{10} > 5$), important to dark orange ($3 < \log B^{10} \leq 5$), positive to light orange ($1 < \log B^{10} \leq 3$), and negligible to yellow ($0 < \log B^{10} \leq 1$), coded according to Kass and Raftery (1995). Blobs correspond to B^{10}_{vc} and the heat bar to B^{10}_{v} defined in (2.18).

2.4.2 Iris Data

In order to compare the Fisher linear discriminant and the naive Bayes classifiers with our proposed approach, we use the well-known iris data (Fisher, 1936). The data consist of 50 samples from three iris flower species: setosa, versicolor, and virginica. Four variables are measured for each sample, the length and width of the sepal and petal. Using the four variables we aim to assign a new observed flower to one of the categories. We use the first 40 observations of each category as training data to derive the classification rule and the remaining 10 samples are used to test the rule. This gives a total of 120 training observations and 30 test observations. The data are shown using variables in Figure 2.10 and using linear discriminant axes in Figure 2.11. As the figures show, the training and the test data are reasonably separated on measured variables and on the two linear discriminant axes.

In order to implement the Gaussian effects model with $q = 1$, the median of each variable of the iris data is subtracted before fitting. However, because the data produce only 12 variable-class combinations, the model parameters are difficult to estimate. Hence we fix $p = 1$, that is, believing all variable-class combinations are useful, and $\sigma_\eta^2 = 0$, meaning there is no experimental error layer. The other parameters are estimated by maximising the likelihood (2.3), giving $\hat{\sigma}^2 = 0.161\ (0.011), \hat{\sigma}_\theta^2 = 1.072\ (0.439)$, and $\hat{\mu} = -0.127\ (0.299)$. Classification with our approach yields misclassification of one of the test observations, but the naive Bayes and the linear discriminant classify all 30 test observations correctly, see Figure 2.11 (right panel). The posterior classification probabilities for the misclassified observation, assuming uniform discrete prior, are reported in Table 2.5. The observation is classified wrongly using our proposed approach, but is also close to the correct class. The other methods classify the observation correctly with a large probability. Methods that tend to classify with certainty often easily misclassify other new observations with certainty too. In other words having very certain classification probabilities might be a consequence of overfitting. We see this effect after adding noise variables to the data as follows.

In order to compare the linear discriminant, the naive Bayes, and our proposed approach on the iris data in a low-sample-size-high-dimension setting,

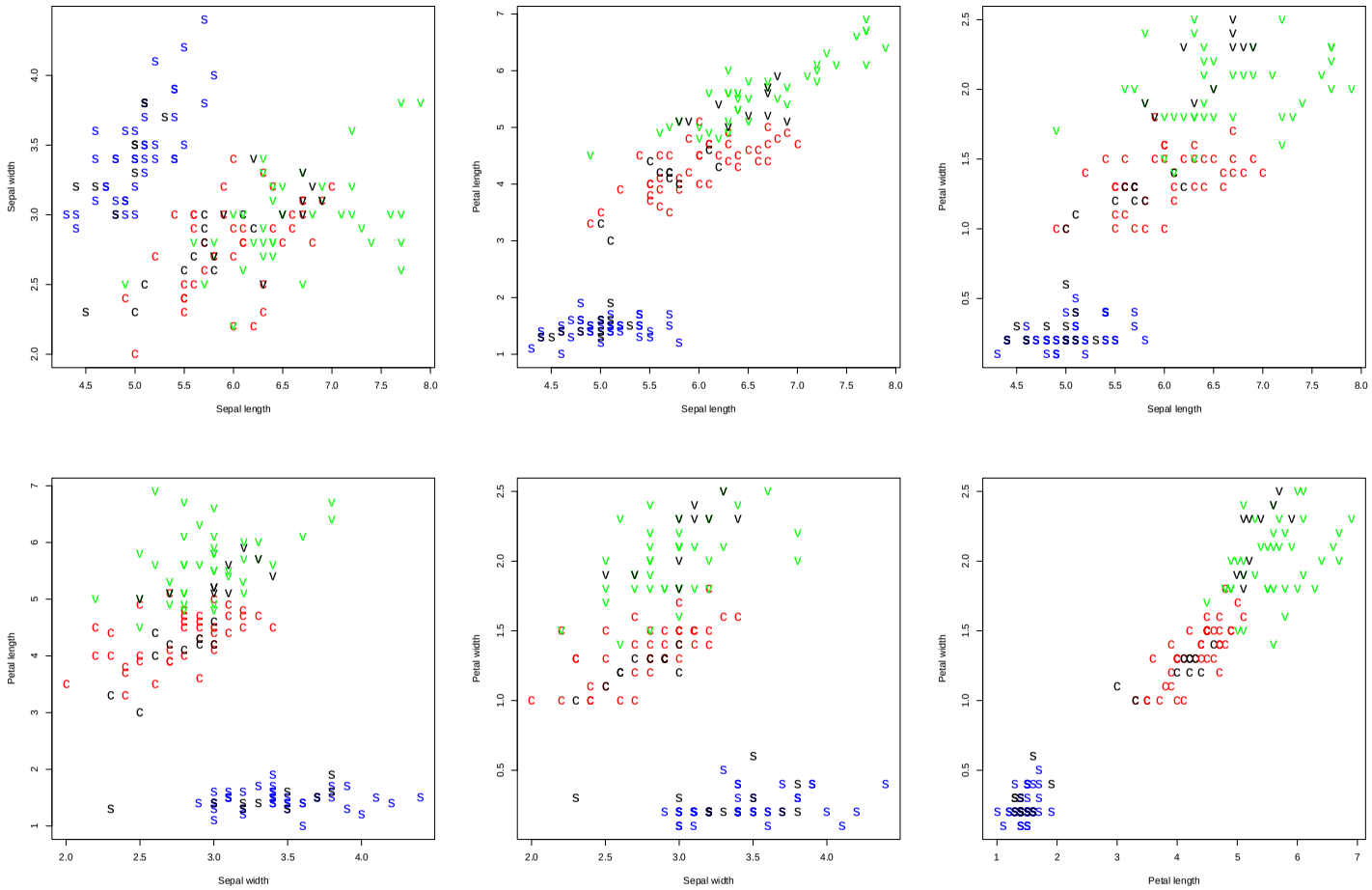

Figure 2.10: Bivariate plots of the iris data. Blue, red and green are used to show the training samples and black to show the test data. Symbols “s”, “c”, and “v” represent setosa, versicolor, and virginica, respectively.

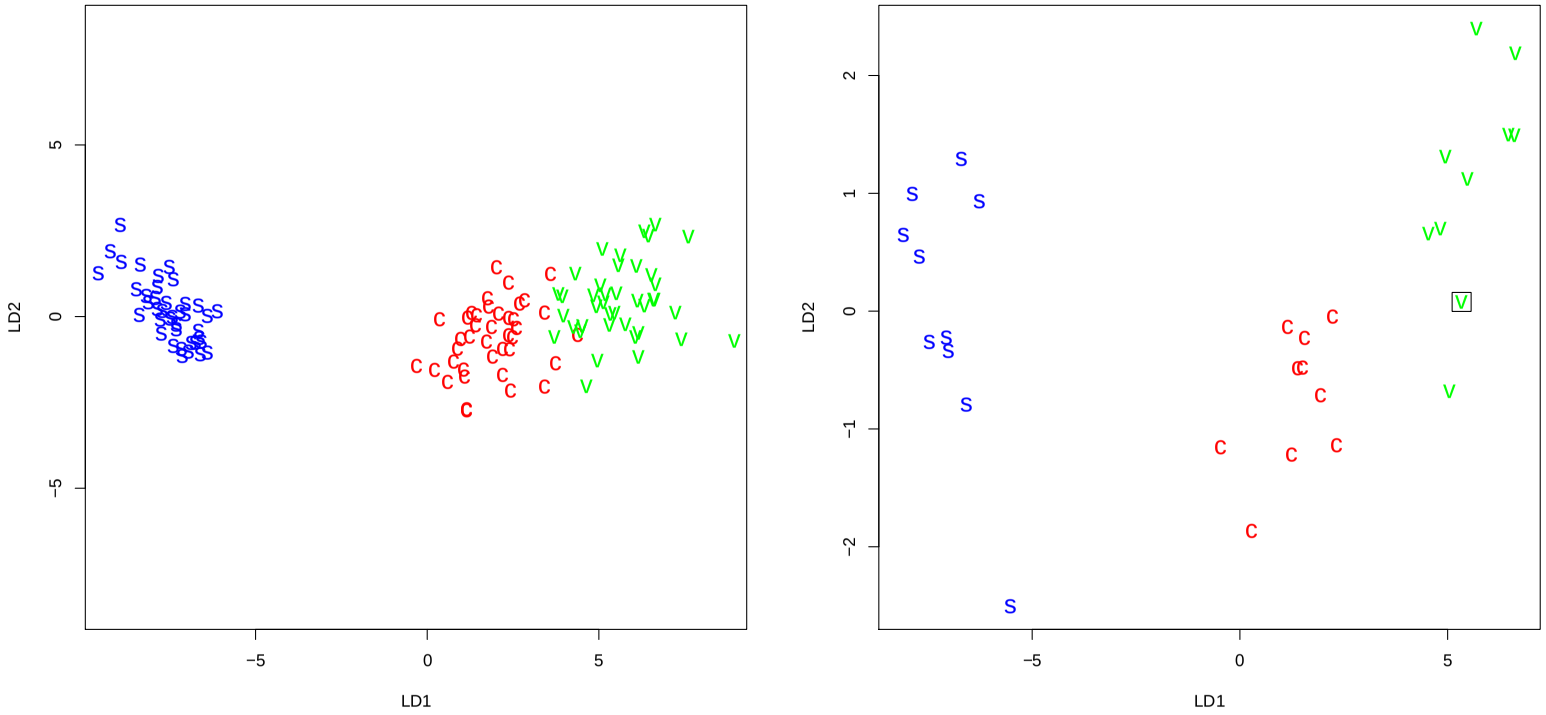

Figure 2.11: The iris data represented on the linear discriminant axes, the training data (left panel) and the test data (right panel). In the right panel, a square shows misclassification with respect to the Gaussian effects model; see the caption to Figure 2.10.

	setosa	*versicolor*	*virginica*
Naive Bayes	0	10.9	89.1
Linear Discriminant	0	2.3	97.7
Gaussian Effects Model	0	50.4	49.6

Table 2.5: Posterior classification percentages assuming a uniform prior for the misclassified virginica subject of the iris data in the right panel of Figure 2.11.

	setosa	*versicolor*	*virginica*
Naive Bayes	0	2.3	97.7
Linear Discriminant	0	100	0
Gaussian Effects Model	0	48.3	51.6

Table 2.6: Posterior classification percentages assuming a uniform prior for the misclassified virginica subject of the Gaussian effects model after adding 196 standard Gaussian noise variables, compare wih Table 2.5.

we add 196 independent standard Gaussian noise variables, yielding a dataset with 200 variables and 150 observations. The Gaussian method is applied after fixing $q = 1$ and estimating the other parameters using maximum likelihood, giving $\hat{\sigma}^2 = 0.986$ (0.009), $\hat{\sigma}^2_\eta = 0.001$ (0.001), $\hat{\sigma}^2_\theta = 1.324$ (0.682), $\hat{\mu} = 0.007$ (0.007), and $\hat{p} = 0.015$ (0.006). The posterior classification probabilities are reported in Table 2.5, confirming that the naive Bayes and the linear discriminant are over-confident, while our approach gives a more stable result. In Table 2.6 we expect probabilities similar to Table 2.6, because in 196 variables the iris samples follow similar patterns, namely standard Gaussian noise. The Gaussian effects model yields one misclassified observation for the test data, which is not the same observation as in the right panel of Figure 2.11, the naive Bayes misclassifies two, and the linear discriminant mistakenly classifies four subjects; see Figure 2.12.

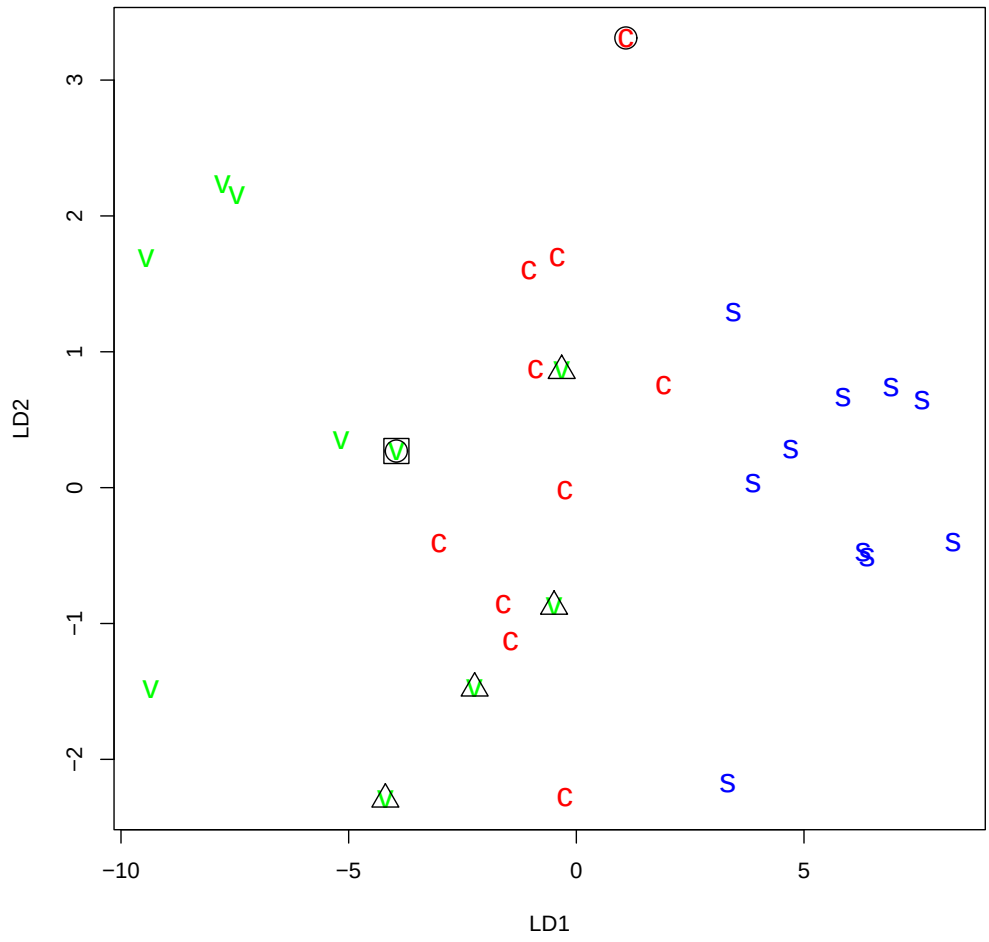

Figure 2.12: The iris data represented on the linear discriminant axes after adding 196 standard Gaussian noise variables. The misclassified observations are shown using a square for the Gaussian effects model, a triangle for the linear discriminant, and a circle for naive Bayes; see the caption to Figure 2.10.

2.5 Analytical Calculations

2.5.1 Introduction

In order to implement classification and clustering the models introduced in this chapter, the calculation of the joint density of the data is required. The joint density, $f(y)$, is calculated in Section 2.5.2 and consists of two parts: the joint density of data in variable v when the variable is inactive, $f(y_v \mid \delta_v = 0)$, which is evaluated in Section 2.5.3, and when the variable is active, $f(y_v \mid \delta_v = 1)$, which is obtained in Section 2.5.4. However, calculation of the joint density for the asymmetric Laplace model requires evaluation of a multivariate integral with integrable function a product of a multivariate Gaussian density and a univariate Gaussian cumulative distribution. This is calculated in Section 2.5.5. For estimation of the model parameters evaluation of the likelihood is also required, calculated in Section 2.5.6. It is easier to do the calculations using the hierarchical model (2.2). For simplicity we denote η'_{vct} by η_{vct}, and θ'_{vc} by θ_{vc}.

2.5.2 Joint Density

The joint density of data plays a key role in model-based classification, because the joint density can be regarded as a distance used to classify a new observation. The new observation is classified to the class having the largest joint density.

Assume that the data have C classes, each consisting of T_c types, and that each type has R_{ct} replicates measured on V variables. Since the models impose independent variables, we can write the overall density density for the data in terms of the densities for observations on the variables, y_v, as

$$f(y) = \prod_{v=1}^{V} f(y_v),$$

and by conditioning on the Bernoulli variable δ_v we can write

$$f(y) = \prod_{v=1}^{V} \left\{ q f(y_v \mid \delta_v = 1) + (1-q) f(y_v \mid \delta_v = 0) \right\}, \tag{2.19}$$

but when $\delta_v = 0$ no variable-class combination is active, yielding

$$f(y_v \mid \delta_v = 0) = \prod_{c=1}^{C} \prod_{t=1}^{T_c} f_0(y_{vct}),$$

where $f_0(y_{vct}) = f(y_{vct} \mid \delta_v = 1, \gamma_{vc} = 0)$; for details see Section 2.5.3. However, for active variables, data in different classes only are independent, that is

$$f(y_v \mid \delta_v = 1) = \prod_{c=1}^{C} f(y_{vc} \mid \delta_v = 1). \tag{2.20}$$

By summing over values of the Bernoulli variable γ_{vc} we may write

$$f(y_{vc} \mid \delta_v = 1) = pf(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) + (1-p)f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 0).$$

When $\gamma_{vc} = 0$, that is variable-class combination for variable v and class c is inactive, the types inside the class become independent, and hence

$$f(y_{vc} \mid \delta_v = 1) = pf_1(y_{vc}) + (1-p) \prod_{t=1}^{T_c} f_0(y_{vct}), \tag{2.21}$$

where, $f_1(y_{vc}) = f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$ is the density of data of variable v and class c, sharing the same θ_{vc}, but involving types with different values of η_{vct}, $t = 1, \ldots, T_c$. The density $f_1(y_{vc})$ is different for Gaussian and asymmetric Laplace models, and is calculated in Section 2.5.4. The full joint density is obtained by inserting these expressions into (2.19).

2.5.3 Density for Inactive Variables

Calculation of the density for inactive variables is easy because when the variable is inactive, all variable-class combinations for that variable are also inactive. Hence, the Gaussian and the asymmetric Laplace models for the inactive variable v reduce to the same model in which θ_{vc} disappears for all $c = 1, \ldots, C$, and

$$y_{vctr} \mid \eta_{vct} \overset{\text{iid}}{\sim} N(\eta_{vct}, \sigma^2), \quad \eta_{vct} \overset{\text{iid}}{\sim} N(\mu, \sigma_\eta^2). \tag{2.22}$$

According to the reduced model (2.22), we can write

$$f(y_v \mid \delta_v = 0) = \prod_{c=1}^{C} \prod_{t=1}^{T_c} f_0(y_{vct}), \tag{2.23}$$

where $f_0(y_{vct}) = f(y_{vct} \mid \delta_v = 1, \gamma_{vc} = 0)$ is the joint density of replications when the variable-class combination is inactive; for the notation see page 2.2.1. One may evaluate f_0 as

$$\begin{aligned} f_0(y_{vct}) &= \int_{-\infty}^{\infty} \prod_{r=1}^{R_{ct}} f(y_{vctr} \mid \eta_{vct}) f(\eta_{vct}) d\eta_{vct} \\ &= (2\pi\sigma^2)^{-R_{ct}/2} (2\pi\sigma_\eta^2)^{-1/2} \\ &\quad \times \int_{-\infty}^{\infty} \exp\left[-\frac{1}{2\sigma^2} \left\{ \sum_{r=1}^{R_{ct}} (y_{vctr} - \eta_{vct})^2 \right\} - \frac{1}{2\sigma_\eta^2} (\eta_{vct} - \mu)^2 \right] d\eta_{vct}. \end{aligned}$$

After completing the square of η_{vct} inside the exponent function, we have a univariate Gaussian integral. Algebraic simplification yields

$$\begin{aligned} f_0(y_{vct}) &= (2\pi)^{-R_{ct}/2} \sigma^{1-R_{ct}} (R_{ct}\sigma_\eta^2 + \sigma^2)^{-1/2} \\ &\quad \times \exp\left\{ -\frac{1}{2\sigma^2} \left(\sum_{r=1}^{R_{ct}} y_{vctr}^2 - R_{ct} \overline{y}_{vct}^2 \right) - \frac{(\overline{y}_{vct} - \mu)^2}{2(\sigma_\eta^2 + \sigma^2/R_{ct})} \right\}, \end{aligned} \tag{2.24}$$

where $\overline{y}_{vct} = R_{ct}^{-1} \sum_{r=1}^{R_{ct}} y_{vctr}$. Replacing (2.24) in (2.23) gives the joint density for the inactive variable v.

2.5.4 Density for Active Variables

Gaussian Model

In order to evaluate the joint density for active variable v, $f(y_v \mid \delta_v = 1)$, according to (2.20) and (2.21) it is required only to evaluate the joint density of data in variable v and class c when the variable and the variable-class combination is active, that is $f_1(y_{vc}) = f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$. When the variable v and variable-class combination in class c are active, the Gaussian effects model reduces to

$$\begin{aligned} y_{vctr} &\mid \eta_{vct} \overset{\text{iid}}{\sim} N(\eta_{vct}, \sigma^2), \\ \eta_{vct} &\mid \theta_{vc} \overset{\text{iid}}{\sim} N(\theta_{vc}, \sigma_\eta^2), \\ &\quad \theta_{vc} \overset{\text{iid}}{\sim} N(\mu, \sigma_\theta^2). \end{aligned} \tag{2.25}$$

Assuming η_{vc} is a vector of length T_c with elements η_{vct} and $\mathbf{Z}$ is a convenient design matrix having $\sum_{t=1}^{T_c} R_{ct}$ rows and T_c columns, we may re-express the reduced model (2.25) as

$$y_{vc} \mid \eta_{vc} \sim N_{\sum_{t=1}^{T_c} R_{ct}} \left(\mu + \mathbf{Z}\eta_{vc}, \sigma^2\mathbf{I}\right), \quad \eta_{vc} \sim N_{T_c}(\mathbf{0}, \mathbf{\Omega}),$$

where, N_p represents a p-variate Gaussian distribution. The covariance matrix $\mathbf{\Omega}_{T_c \times T_c}$ is a uniform covariance matrix having main diagonals $\sigma_\eta^2 + \sigma_\theta^2$ and off-diagonals σ_θ^2 obtained after integration over a univariate θ_{vc}. Hence, using standard mixed effects calculations (McCulloch and Searle, 2001, p. 159) we have

$$y_{vc} \sim N_{\sum_{t=1}^{T_c} R_{ct}}(\mu\mathbf{1}, \mathbf{\Sigma} = \sigma^2\mathbf{I} + \mathbf{Z}\mathbf{\Omega}\mathbf{Z}^{\mathrm{T}}).$$

The covariance matrix $\mathbf{\Sigma}$ corresponds to a $\sum_{t=1}^{T_c} R_{ct} \times \sum_{t=1}^{T_c} R_{ct}$ matrix, with $\sigma^2 + \sigma_\eta^2 + \sigma_\theta^2$ on the main diagonals, off-diagonals equal to $\sigma_\eta^2 + \sigma_\theta^2$ for replications of the same type, and to σ_θ^2 for observations emerging from different types. For example assume vector y_{vc} consisting of univariate y_{vctr}, includes two types; one with two replications and another with three replications. Hence

$$y_{vc} = \begin{pmatrix} y_{vc11} \\ y_{vc12} \\ y_{vc21} \\ y_{vc22} \\ y_{vc23} \end{pmatrix},$$

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma^2 + \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\theta^2 & \sigma_\theta^2 & \sigma_\theta^2 \\ \sigma_\eta^2 + \sigma_\theta^2 & \sigma^2 + \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\theta^2 & \sigma_\theta^2 & \sigma_\theta^2 \\ \sigma_\theta^2 & \sigma_\theta^2 & \sigma^2 + \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 \\ \sigma_\theta^2 & \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 & \sigma^2 + \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 \\ \sigma_\theta^2 & \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 & \sigma_\eta^2 + \sigma_\theta^2 & \sigma^2 + \sigma_\eta^2 + \sigma_\theta^2 \end{pmatrix}.$$

Asymmetric Laplace Model

Calculation of the joint density of the asymmetric Laplace model when variable v is active is similar to the Gaussian case. First we calculate the multivariate density for vector η_{vc} comprising of elements η_{vct}. When the variable

and the variable-class combination are active, the density $f(\eta_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$ equals

$$\int_{-\infty}^{\infty} f(\eta_{vc} \mid \theta_{vc}, \delta_v = 1, \gamma_{vc} = 1) f(\theta_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) d\theta_{vc}$$

and this equals

$$\int_{-\infty}^{\mu} (2\sigma_{\theta_L})^{-1} (2\pi\sigma_\eta^2)^{-T_c/2} \exp\left\{ -\frac{1}{2\sigma_\eta^2} \sum_{t=1}^{T_c} (\eta_{vct} - \theta_{vc})^2 + \frac{\theta_{vc} - \mu}{\sigma_{\theta_L}} \right\} d\theta_{vc}$$
$$+ \int_{\mu}^{+\infty} (2\sigma_{\theta_R})^{-1} (2\pi\sigma_\eta^2)^{-T_c/2} \exp\left\{ -\frac{1}{2\sigma_\eta^2} \sum_{t=1}^{T_c} (\eta_{vct} - \theta_{vc})^2 + \frac{\mu - \theta_{vc}}{\sigma_{\theta_R}} \right\} d\theta_{vc}$$

or

$$(2\pi\sigma_\eta^2)^{-T_c/2} \left\{ (2\sigma_{\theta_L})^{-1} I_{1L} + (2\sigma_{\theta_R})^{-1} I_{1R} \right\}, \tag{2.26}$$

where

$$\begin{aligned} I_{1L} &= \exp\left\{ -\sum_{t=1}^{T_c} \eta_{vct}^2 - \frac{\mu}{\sigma_{\theta_L}} + \frac{T_c}{2\sigma_\eta^2} \left(\overline{\eta}_{vc} + \frac{\sigma_\eta^2}{T_c \sigma_{\theta_L}} \right)^2 \right\} \\ &\quad \times \int_{-\infty}^{\mu} \exp\left[-\frac{T_c}{2\sigma_\eta^2} \left\{ \theta_{vc} - \left(\overline{\eta}_{vc} + \frac{\sigma_\eta^2}{T_c \sigma_{\theta_L}} \right) \right\}^2 \right] d\theta_{vc}, \end{aligned}$$

in which $\overline{\eta}_{vc} = T_c^{-1} \sum_{t=1}^{T_c} \eta_{vct}$. Letting Φ denote the standard Gaussian cumulative distribution function, the last integral equals

$$(2\pi\sigma_\eta^2/T_c)^{1/2} \Phi\left\{ \frac{\mu - \overline{\eta}_{vc} - \sigma_\eta^2/(T_c \sigma_{\theta_L})}{\sqrt{\sigma_\eta^2/T_c}} \right\}.$$

Similarly,

$$\begin{aligned} I_{1R} &= \exp\left\{ -\sum_{t=1}^{T_c} \eta_{vct}^2 + \frac{\mu}{\sigma_{\theta_R}} + \frac{T_c}{2\sigma_\eta^2} \left(\overline{\eta}_{vc} - \frac{\sigma_\eta^2}{T_c \sigma_{\theta_R}} \right)^2 \right\} \\ &\quad \times (2\pi\sigma_\eta^2/T_c)^{1/2} \Phi\left\{ \frac{\overline{\eta}_{vc} - \mu - \sigma_\eta^2/(T_c \sigma_{\theta_R})}{\sqrt{\sigma_\eta^2/T_c}} \right\}. \end{aligned}$$

Hence, $f_1(y_{vc}) = f(\eta_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$ can be obtained numerically by replacing I_{1L} and I_{1R} in (2.26).

The joint density $f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$, in which y_{vc} is vector of length $\sum_{t=1}^{T_c} R_{ct}$, is obtained by marginalizing over $f(\eta_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$. Hence the density $f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$ equals

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(y_{vc} \mid \eta_{vc}, \delta_v = 1, \gamma_{vc} = 1) f(\eta_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) d\eta_{vc}$$

or

$$(2\pi\sigma^2)^{-\sum_{t=1}^{T_c} R_{ct}/2} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{T_c} \sum_{r=1}^{R_{ct}} (y_{vctr} - \eta_{vct})^2 \right\} \times f(\eta_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) d\eta_{vc},$$

and this equals

$$(2\pi\sigma^2)^{-\sum_{t=1}^{T_c} R_{ct}/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{t=1}^{T_c} \sum_{r=1}^{R_{ct}} y_{vctr}^2 \right) (k_L I_{2L} + k_R I_{2R}),$$

in which

$$\begin{aligned} k_L &= (2\sigma_{\theta_L})^{-1} \exp\left(\frac{\sigma_\eta^2}{2T_c\sigma_{\theta_L}^2} - \frac{\mu}{\sigma_{\theta_L}} \right), \\ k_R &= (2\sigma_{\theta_R})^{-1} \exp\left(\frac{\sigma_\eta^2}{2T_c\sigma_{\theta_R}^2} + \frac{\mu}{\sigma_{\theta_R}} \right), \end{aligned}$$

and

$$\begin{aligned} I_{2L} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} & \exp\left\{ -\frac{1}{2\sigma^2} \left(\sum_{t=1}^{T_c} R_{ct}\eta_{vct}^2 - 2\sum_{t=1}^{T_c} R_{ct}\overline{y}_{vct}\eta_{vct} \right) \right\} \\ & \times \exp\left(\frac{\sum_{t=1}^{T_c} \eta_{vct}^2}{2\sigma_\eta^2} + \frac{T_c\overline{\eta}_{vc}^2}{2\sigma_\eta^2} + \frac{\overline{\eta}_{vc}}{\sigma_{\theta_L}} \right) \\ & \times \Phi\left\{ \frac{\mu - \overline{\eta}_{vc} - \sigma_\eta^2/(T_c\sigma_{\theta_L})}{\sqrt{\sigma_\eta^2/T_c}} \right\} d\eta_{vc}. \end{aligned}$$

The terms inside the exponent functions can be re-arranged as

$$\exp\left\{-\frac{1}{2}\sum_{t=1}^{T_c}\left(\frac{R_{ct}}{\sigma^2}+\frac{1}{\sigma_\eta^2}-\frac{1}{T_c\sigma_\eta^2}\right)\eta_{vct}^2+\sum_{t\neq t'}-\frac{1}{T_c\sigma_\eta^2}\eta_{vct}\eta_{vct'}\right\}\times$$
$$\exp\left\{-2\sum_{t=1}^{T_c}\left(\frac{R_{ct}\overline{y}_{vct}}{\sigma^2}+\frac{1}{T_c\sigma_{\theta_L}}\right)\eta_{vct}\right\},$$

which can be written in a quadratic form using matrix notation as follows. Suppose the vector $\mathbf{b}_L$ of length T_c is composed of elements

$$\frac{R_{ct}\overline{y}_{vct}}{\sigma^2}+\frac{1}{T_c\sigma_{\theta_L}},$$

and consider a square matrix $\mathbf{A}$ with diagonals

$$\frac{R_{ct}}{\sigma^2}+\frac{1}{\sigma_\eta^2}-\frac{1}{T_c\sigma_\eta^2},$$

and equal off-diagonals $-1/T_c\sigma_\eta^2$. Since we have

$$\eta_{vc}^{\mathrm{T}}\mathbf{A}\eta_{vc}-2\mathbf{b}_L^{\mathrm{T}}\eta_{vc}=(\eta_{vc}-\mathbf{A}^{-1}\mathbf{b}_L)^{\mathrm{T}}\mathbf{A}(\eta_{vc}-\mathbf{A}^{-1}\mathbf{b}_L)-\mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L,$$

we can write I_{2L} in a quadratic form as

$$I_{2L}=\exp\left(\frac{1}{2}\mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L\right)\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\exp\left\{-\frac{1}{2}(\eta_{vc}-\mathbf{A}^{-1}\mathbf{b}_L)^{\mathrm{T}}\mathbf{A}(\eta_{vc}-\mathbf{A}^{-1}\mathbf{b}_L)\right\}$$
$$\times\Phi\left(c_L+\mathbf{d}_L^{\mathrm{T}}\eta_{vc}\right)d\eta_{vc}, \tag{2.27}$$

where $\mathbf{d}_L$ is a vector of length T_c with equal elements $-1/\sqrt{T_c\sigma_\eta^2}$ and c_L is a constant being $\{\mu-\sigma_\eta^2/(T_c\sigma_{\theta_L})\}/\sqrt{\sigma_\eta^2/T_c}$. It is easy to verify that the matrix $\mathbf{A}_{T_c\times T_c}$ is positive definite (Rencher, 1998, p. 413). Using the result of Section 2.5.5 we can analytically evaluate the last integral and write

$$I_{2L}=\exp\left(\frac{1}{2}\mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L\right)(2\pi)^{T_c/2}|\mathbf{A}|^{-1/2}\Phi\left(\frac{c_L+\mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L}{\sqrt{1+\mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_L}}\right),$$

in which $|\mathbf{A}|$ denotes the determinant of matrix $\mathbf{A}$.

Similarly we can evaluate

$$I_{2R}=\exp\left(\frac{1}{2}\mathbf{b}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R\right)(2\pi)^{T_c/2}|\mathbf{A}|^{-1/2}\Phi\left(\frac{c_R+\mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R}{\sqrt{1+\mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_R}}\right),$$

where $\mathbf{b}_R$ is a vector of length T_c made of elements $R_{ct}\overline{y}_{vct}/\sigma^2 - 1/(T_c\sigma_{\theta_R})$, $\mathbf{d}_R$ having the same length as $\mathbf{b}_R$, with equal elements $1/\sqrt{T_c\sigma_\eta^2}$ and c_R is a constant being $\{-\mu-\sigma_\eta^2/(T_c\sigma_{\theta_L})\}/\sqrt{\sigma_\eta^2/T_c}$. After putting the pieces together and re-arrangement we get

$$f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) \;=\; k_0(k_L I_k + k_R I_R), \tag{2.28}$$

where

$$\begin{aligned}
k_0 &= (2\pi\sigma^2)^{-\sum_{t=1}^{T_c} R_{ct}/2}(2\pi\sigma_\eta^2)^{-T_c/2}(2\pi\sigma_\eta^2/T_c)^{1/2} \times \\
&\quad (2\pi)^{T_c/2}|\mathbf{A}|^{-1/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{r=1}^{R_{ct}}\sum_{t=1}^{T_c} y_{vctr}^2\right\}, \\
k_L &= (2\sigma_{\theta_L})^{-1}\exp\left(\frac{\sigma_\eta^2}{2T_c\sigma_{\theta_L}^2} - \frac{\mu}{\sigma_{\theta_L}}\right), \\
I_L &= \exp\left(\frac{1}{2}\mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L\right)\Phi\left(\frac{c_L + \mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L}{\sqrt{1+\mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_L}}\right), \\
k_R &= (2\sigma_{\theta_R})^{-1}\exp\left(\frac{\sigma_\eta^2}{2T_c\sigma_{\theta_R}^2} + \frac{\mu}{\sigma_{\theta_R}}\right), \\
I_R &= \exp\left(\frac{1}{2}\mathbf{b}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R\right)\Phi\left(\frac{c_R + \mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R}{\sqrt{1+\mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_R}}\right).
\end{aligned}$$

2.5.5 Multivariate Gaussian Density-Distribution Integral

In the analytical calculation of the joint density for the asymmetric Laplace model in (2.27) we encounter an integral having the form

$$\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\exp\left\{-\frac{1}{2}(\eta-\mathbf{A}^{-1}\mathbf{b})^{\mathrm{T}}\mathbf{A}(\eta-\mathbf{A}^{-1}\mathbf{b})\right\}\Phi(c+\mathbf{d}^{\mathrm{T}}\eta)d\eta,$$

in which η, $\mathbf{b}$ and $\mathbf{d}$ are vectors of length p, $\mathbf{A}$ is a $p\times p$ positive definite matrix, and c is a constant. We may rewrite the integral above in terms of Gaussian density-distribution as follows

$$(2\pi)^{p/2}|\mathbf{A}|^{-1/2}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\phi_*(\eta)\Phi(c+\mathbf{d}^{\mathrm{T}}\eta)d\eta,$$

where ϕ_* denotes the multivariate Gaussian density with mean $\mathbf{A}^{-1}\mathbf{b}$ and covariance matrix $\mathbf{A}^{-1}$. We may re-express the last multivariate integral in terms of a univariate Gaussian random variable Z as $\Pr(Z - \mathbf{d}^{\mathrm{T}}\eta < c)$. Now we define $\eta^* = (Z, \eta)^{\mathrm{T}}$, $\mathbf{d}^* = (1, -\mathbf{d}^{\mathrm{T}})^{\mathrm{T}}$ and a univariate Gaussian random variable $Z^* = \mathbf{d}^{*\mathrm{T}}\eta^*$. Re-expressing the integral using Z^* which has mean $-\mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}$, and variance $1 + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}$ gives the analytic solution as

$$(2\pi)^{p/2}|\mathbf{A}|^{-1/2} \Pr\left(\frac{Z^* + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}{\sqrt{1 + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}}} < \frac{c + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}{\sqrt{1 + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}}}\right),$$

or

$$(2\pi)^{p/2}|\mathbf{A}|^{-1/2}\Phi\left(\frac{c + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}{\sqrt{1 + \mathbf{d}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}}}\right). \tag{2.29}$$

2.5.6 Likelihood

Gaussian Model

The likelihood of data used to estimate the model parameters is the joint density under the assumption that each class consists of a single type, that is $T_c = 1$ for all $c = 1, \ldots, C$; hence we drop the index t. The likelihood can be calculated in the same way as the joint density, except that η_{vc} is univariate. We marginalize first on δ_v, and then on γ_{vc}, yielding

$$f(y) = \prod_{v=1}^{V} f(y_v) = \prod_{v=1}^{V} \{qf(y_v \mid \delta_v = 1) + (1 - q)f(y_v \mid \delta_v = 0)\},$$

in which $f(y_{vc} \mid \delta_v = 0)$ allows no variable-class combination to be active, and simplifies to

$$f(y_{vc} \mid \delta_v = 0) = \prod_{c=1}^{C} f_0(y_{vc}).$$

However, when the variable is active, the appearance of the true effect depends on γ_{vc}, yielding

$$f(y_v \mid \delta_v = 1) = pf(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1) + (1 - p)f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 0),$$

where $f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 0) = f_0(y_{vc})$ is obtained by integration over a univariate η_{vc} and is already calculated in (2.24). The density $f(y_{vc} \mid \delta_v =$

$1, \gamma_{vc} = 1) = f_1(y_{vc})$ can be calculated similarly by first integrating η_{vc} over a univariate θ_{vc} and then integrating y_{vc} over the marginalized univariate η_{vc}. This gives a density having a similar form as $f_0(y_{vc})$ but with σ_η^2 replaced with $\sigma_\eta^2 + \sigma_\theta^2$.

Asymmetric Laplace Model

Derivation of the likelihood is similar to the Gaussian model, but $f_1(y_{vc}) = f(y_{vc} \mid \delta_v = 1, \gamma_{vc} = 1)$ which is calculated (2.28), must be obtained by integrating over a univariate η_{vc} with $T_c = 1$ $(c = 1, \ldots, C)$. This means $\mathbf{b}_L, \mathbf{d}_L, \mathbf{b}_R, \mathbf{d}_R$ are vectors of length one and matrix $\mathbf{A}$ is a 1×1 matrix in (2.28). Hence, inverse of $\mathbf{A}$ is inverse of its single element and $|\mathbf{A}|$ equals its single element. Re-arrangement and simplifying the result yields (2.10).

Chapter 3

Clustering

3.1 Introduction

3.1.1 General

The goal of cluster analysis is to partition observations into groups such that the observations belonging to the same group are more similar than observations belonging to different groups. There are various ways of attributing observations to different clusters but one may classify clustering methods into two categories, distance-based and model-based techniques. Our approach, as described below, is in-between, because we use a model to define a distance and we implement agglomerative clustering as used in distance-based methods.

Some preliminaries are given in this section. In Section 3.2 agglomerative clustering is presented using the Gaussian and the asymmetric Laplace variable selection models. Computational issues related to our proposed clustering method are discussed in Section 3.3 and our computer code is briefly analysed. Section 3.4 shows the application of our technique for analysis of the metabolite, microarray, and the image data. The advantages and some disadvantages of our approach are briefly presented in Section 3.5.

In distance-based methods a distance or dissimilarity measure between groups of observations is often defined and a reasonable objective function is optimised to obtain a grouping. A very common algorithm is the k-means

approach which we briefly describe below.

Suppose y_{vt} denotes observation t $(t = 1, \ldots, T)$ measured on variable v $(v = 1, \ldots, V)$. For convenience assume that y_{vt} involves a single observation. One way of defining a dissimilarity measure between a pair of observations y_t and $y_{t'}$ $(t \neq t')$ is by taking

$$\begin{aligned} S(y_t, y_{t'}) &= \sum_{v=1}^{V} w_v s(y_{vt}, y_{vt'}), \\ w_v &\geq 0, \\ \sum_{v=1}^{V} w_v &= 1, \end{aligned} \tag{3.1}$$

a convex combination of s, the distance between pairs of observations for each variable. The weights w_v are subjective and chosen for each variable v. A common choice is a constant $w_v = 1/V$, but assigning equal weights does not mean that variables have equal influence. The influence of vth variable depends upon its relative contribution over all pairs of observation. The choice of distance between observations is also optional, but often squared difference is chosen

$$s(y_{vt}, y_{vt'}) = (y_{vt} - y_{vt'})^2. \tag{3.2}$$

Each observation is assigned to a cluster using an integer label. We denote a vector of length T of such labels by $\mathbf{d}$, consisting of positive integer elements d_t $(t = 1, \ldots, T)$. Individuals in the same group take the same value in $\mathbf{d}$. For example consider five observations with $\mathbf{d} = (1, 1, 2, 1, 1)$. This denotes data forming two clusters, in which the third individual makes his own cluster and the others are in the same group. However, $\mathbf{d} = (2, 2, 1, 2, 2)$ refers also to the same grouping. In order to make the labelling unique, we consider that the first individual always takes integer 1, the second individual takes label 1 if it is in the same cluster as the first individual and takes 2 otherwise, and so forth. Hence the labelling vector $\mathbf{d} = (2, 2, 1, 2, 2)$ never appears. The tth observation belongs to the class c if the tth element of $\mathbf{d}$ equals c i.e. $d_t = c$, $(c = 1, \ldots, C)$. The maximum number of non-empty clusters C is T and the minimum number is 1, so $C \in \{1, \ldots, T\}$.

In order to cluster data an objective function is also required. This may be defined following the analysis of variance idea for the squared difference distance, that is for $t \neq t'$,

$$S_T = S_W + S_B, \tag{3.3}$$

in which

$$\begin{aligned} S_W &= \sum_{c=1}^{C} \sum_{\{t|d_t=c\}} \sum_{\{t'|d_{t'}=c\}} S(y_t, y_{t'}), \\ S_B &= \sum_{c=1}^{C} \sum_{\{t|d_t=c\}} \sum_{\{t'|d_{t'}\neq c\}} S(y_t, y_{t'}) \\ S_T &= \sum_{t\neq t'} S(y_t, y_{t'}). \end{aligned}$$

The total dissimilarity S_T does not depend on data grouping, but the within-cluster dissimilarity S_W and the between-cluster dissimilarity S_B, both depend on data allocation. Hence according to (3.3) maximising S_B is equivalent to minimising S_W. However, direct optimisation of S_B or S_W is not easy and often iterative algorithms are used.

The number of groupings of T observations into C clusters is the Stirling number of the second kind (Jain and Dubes, 1988),

$$B(T, C) = \frac{1}{C!} \sum_{c=1}^{C} (-1)^{C-c} \begin{pmatrix} C \\ c \end{pmatrix} c^T,$$

which equals 34105 for $C = 4$ and $T = 10$, and about 10^{10} for $C = 4$ and $T = 19$. Hence, the number of partitions of a set, which is known to be the Bell number, equals

$$B(T) = \sum_{C=1}^{T} B(T, C).$$

Clustering is a difficult problem, since $B(T, C)$ is a large number, and the problem becomes even more complicated when the number clusters C is also unknown, because the search space $B(T)$ grows extremely rapidly with increasing T. For instance $B(40) = 1.6 \times 10^{35}$ and $B(100) = 4.8 \times 10^{115}$.

Hence, with a moderate sample size, no optimisation routine can visit more than a tiny fraction of all possible allocations.

One of the most common optimisation algorithms for clustering using the dissimilarity defined in (3.2) is the k-means algorithm. This uses a fixed number of clusters C and minimises S_W. One may write

$$S_W = \sum_{c=1}^{C} T_c \sum_{\{t|d_t=c\}} \sum_{v=1}^{V} (y_{vct} - \overline{y}_{vc})^2,$$

where T_c denotes the number of observations in cluster c, and $\overline{y}_{vc}$ is the mean of cluster c for variable v. The optimisation works as follows. First, for a given labelling $\mathbf{d}$, a centre for cluster c is chosen such that the within-cluster variance is minimised for each variable, that is minimising $S_W(v,c) = \sum_{\{t|d_t=c\}} \sum_{v=1}^{V} (y_{vct} - m_{vc})^2$, thus yielding $m_{vc} = \overline{y}_{vc}$. Second, the closest points are attributed to each cluster according to S_W. These two steps are iterated until convergence of the objective function.

The k-means algorithm has some disadvantages, for example it may give different answers according to the starting values. In addition, the number of clusters of data, C, is often unknown and it is not very clear how one can choose the best candidate from different candidates $C \in \{1, \ldots, T\}$. One way of choosing C is a visual approach, that is to create a visual guide of different partitioning for different choices of the number of clusters, e.g. creating a dendrogram by taking a special path over various choices of $\mathbf{d}$.

An interesting path is one which gives an ordered set of labels, that is in which a grouping with C clusters is a refinement of a grouping with $C-1$ clusters. The ordered paths of four observations are shown in Figure 3.1. Clustering methods that take the ordered paths are called hierarchical clustering and the trees made by such methods are named dendrogram.

Hierarchical clustering has two variants, agglomerative and divisive. Agglomerative clustering starts initially with each observation as a separate cluster, successively adds the closest cluster using a dissimilarity measure, and continues merging clusters until all observations are in one cluster. In contrast, divisive clustering starts with all observations in one cluster and divides clusters until finishing with each observation as a single cluster. An example of a dendrogram created by agglomerative clustering is shown in

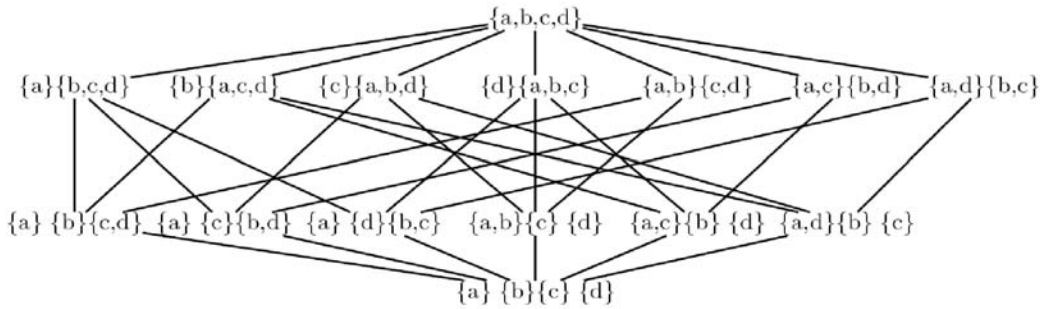

Figure 3.1: All possible partitionings of four observations $\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$. The solid lines represent the partial ordering. Two partitions are ordered if there is a path connecting the two partitions, that is, one partition is a refinement of another.

Figure 3.2.

The choice of dissimilarity measure in hierarchical clustering is arbitrary, like the k-means (3.2). If the dissimilarity is based on the most distant points of the two clusters, the method is called complete linkage, if based on the closest points or the nearest neighbours, it is called single linkage, and if based on the average distances, it is called average linkage.

Complete linkage is more popular because it gives nice bifurcating dendrograms, but the asymptotic behaviour of the complete linkage method depends on the regions that the observations belong to and not on the probability distribution of the data. This happens because after some clustering steps the complete linkage dissimilarity becomes independent of the number of observations in the groups. This is a considerable disadvantage, because from a model-based point of view data in the same cluster share the same probability distribution, which complete linkage appears to discard. For more discussion see Hartigan (1985).

3.1.2 Model-Based Clustering

In model-based clustering a family of statistical models is considered for data, and clustering is implemented by fitting a mixture model. Assume that data in cluster c follow the parametric model $f(y_c \mid \theta_c)$. Then the overall distribution is $\sum_{c=1}^{C} p_c f(y_c \mid \theta_c)$, a mixture model in which p_c is the proportion

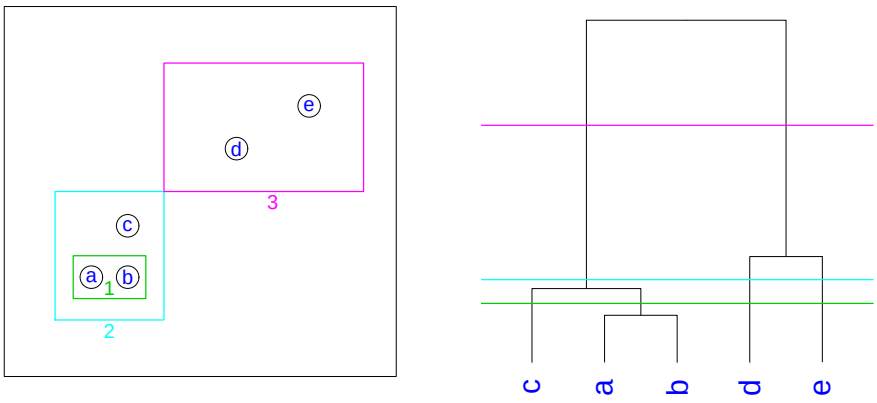

Figure 3.2: Example of five bivariate observations (left panel) with their dendrogram made by an agglomerative method (right panel). The grouping at each clustering step can be obtained by cutting the dendrogram.

of data belonging to cluster c. Often fitting mixtures is implemented using the EM algorithm of Dempster *et al.* (1977), found to be a generalisation of the k-means algorithm (Hastie *et al.*, 2001, p. 505). The parametric family $f(y_c \mid \theta_c)$ is often chosen to be the Gaussian family. However, like the k-means algorithm, applying the EM algorithm is feasible only when the number of clusters is specified, but unlike the k-means algorithm, the model provides a criterion for choosing the number of clusters, like the AIC (Akaike, 1973) or the BIC (Schwarz, 1978).

If T_c observations are in cluster c, then the data distribution is the same if the observations in cluster c are arbitrarily reordered. This means that $f(y_{1c} \ldots y_{T_c c})$ is an exchangeable distribution and by the general representation theorem (Bernardo and Smith, 1994, Chapter 4), there is a conditional density $f(y_c \mid \theta_c)$ and a prior density $f(\theta_c)$ such that

$$f(y_{1c}, \ldots, y_{T_c c}) = \int \prod_{t=1}^{T_c} f(y_{tc} \mid \theta_c) f(\theta_c) d\theta_c.$$

The general representation theorem clearly suggests use of a Bayesian model for clustering problem.

The posterior distribution in Bayesian clustering is complicated and can-

not be easily summarised and maximised. Ordinary Markov chain Monte Carlo samplers are known to be computationally inefficient. Alternative samplers have been proposed in the literature (Jain and Neal, 2004; Dahl, 2003; Jain and Neal, 2007) which all try to explore the search space more effectively by adding clever split-merge moves.

We propose to take the agglomerative clustering path using the marginal posterior density value as the similarity measure (Heller and Ghahramani, 2005) thereby gaining a dendrogram representation. The marginal posterior provides a measure of the plausibility of a merge which is supported by the model if the posterior increases for that merge. This provides a guide to where to cut the resulting dendrogram.

3.1.3 High-Dimensional Clustering

It is well-known in statistical modelling that statistical analyses become difficult in high dimensions. Classification may be regarded as a simplified version of clustering in which the label of only one observation is unknown. Hence, it is not surprising that we encounter overfitting problem in model-based clustering as well. Overfitting appears because of the lack of a valid statistical model when the number of observations is small but the number of variables is large. The problem was discussed for classification in Section 2.1.2.

In order to solve the overfitting problem, data are often projected to a smaller dimension or variables relevant to the analysis are chosen. However, it is hard to define a valid criterion for optimally projecting data without loss of clustering information. Classical projection methods like principal components are not necessarily relevant. Chang (1983) argues that clustering projections may appear in the last principal components, which are often ignored.

Projection pursuit (Friedman and Tukey, 1974) is a more convenient method to capture nonlinear patterns in data and can be regarded as a generalisation of the principal components method (Friedman, 1987). Projection pursuit often optimises a criterion that reflects multi-modality and hence give projections relevant to clustering. Diaconis and Friedman (1984) show that for most high-dimensional data, most low dimensional projections are ap-

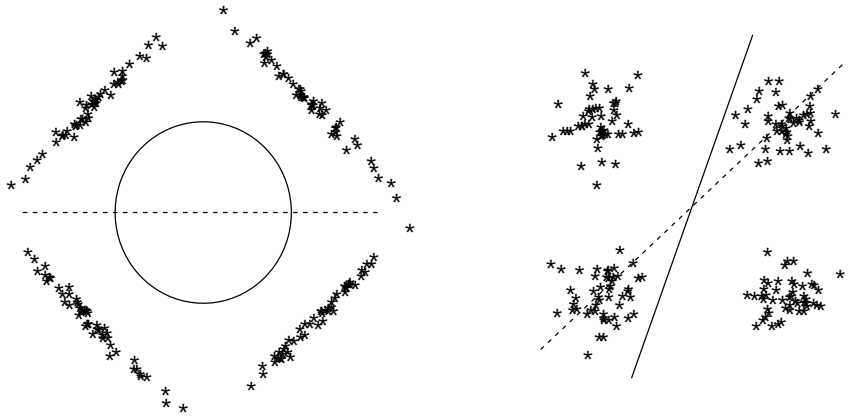

Figure 3.3: Examples where dimension reduction using principal components (dots) yields loss of clustering information. An appropriate lower dimensional projection (solid) is nonlinear in the left panel and is linear in the right panel.

proximately Gaussian, hence one may argue that interesting projections are the ones that are non-Gaussian. Being far from Gaussian cannot however be uniquely defined, Huber (1985) gave a list of reasonable criteria. There are limitations in the use of projection, because highly nonlinear effects cannot be captured with projection pursuit (Jones and Sibson, 1987). Figure 3.3 shows interesting and uninteresting projections of a two-dimensional data cloud, in which dimension reduction with principal components yields loss of the clustering information. Even if a good projection method is available, that is, data points are clustered reasonably well in the projected lower dimensional space, inference about important clustering variables as demanded in many applications, is troublesome.

Another approach to solving the curse of high-dimensionality is variable selection. Selecting variables becomes difficult in clustering because there is no clear response variable to guide the search. A variable is important if it helps to define a mixture, and is unimportant otherwise. Fitting mixtures is hard even with a fixed number of variables and incorporating variable selection based on a vague criterion complicates the matter further.

Researchers have dealt with the curse of dimensionality in clustering in various ways. McLachlan *et al.* (2002) in a microarray data analysis use

forward selection of variables by applying a univariate test of a single component versus a mixture of two components, but this method selects a lot of variables. Wang and Zhu (2008) and Bondell and Reich (2008) implement variable selection using penalised likelihood, but choice of the penalising constant is nontrivial and arbitrary. Friedman and Meulman (2004) assign different weights to groups of variables but the weights can be estimated only for a fixed subset of clustering subjects. Liu *et al.* (2003) select principal components using Gibbs sampler, however it is known from Chang (1983) that principal components may give irrelevant projections. Bensmail *et al.* (2005) denoise data using the Fourier transform and fit Gaussian mixtures on the denoised data, but the choice of denoising threshold is subjective.

Raftery and Dean (2006) applied variable selection using an approximate Bayes factor, but their approach is not appropriate when the number of variables exceeds the sample size. The Bayesian framework enables the fitting of mixtures with unknown numbers of components and variable selection jointly through the reversible jump algorithm of Green (1995). Tadesse *et al.* (2005) have applied variable selection in Bayesian model-based clustering using finite Gaussian mixtures, and Kim *et al.* (2006) have implemented the variable selection for Dirichlet mixture models, both using a trans-dimensional Markov chain Monte Carlo method which is computationally challenging and slow. However, considering T observations and V number of variables, the Markov chain required for such analyses have $2^V B(T)$ states if we allow variable selection jointly with grouping, where $B(T)$ is the Bell number. This is huge for high-dimensional problems with moderate sample size. Hence, even with current computational power a Markov chain cannot visit more than a fraction of the possible states.

3.1.4 Clustering Prior

When a statistical model is assumed, a natural metric is imposed, the joint density of the data, and this criterion can be used to judge about merging or splitting clusters.

In the Bayesian paradigm the posterior increases up to a point as clusters are joined and then decreases, and hence can be used to choose the best

$C=3$				$C=2$			$C=1$	
T_1	T_2	T_3	$f(\mathbf{d})$	T_1	T_2	$f(\mathbf{d})$	T_1	$f(\mathbf{d})$
0	0	3	4.58	0	3	11.45	3	45.80
1	0	2	1.53	1	2	3.82		
2	0	1	1.53	2	1	3.82		
3	0	0	4.58	3	0	11.45		
0	1	2	1.53					
1	1	1	0.76					
2	1	0	1.53					
0	2	1	1.53					
1	2	0	1.53					
0	3	0	4.58					

Table 3.1: Prior clustering probabilities ($\times 100$) for $T = 3$ based on (3.6).

grouping. In order to implement Bayesian clustering, a prior is required for data grouping represented by the vector $\mathbf{d}$.

We assume an exchangeable prior, so it is enough to assume a prior for T_c, the number of observations in cluster c ($c = 1 \ldots, C$), and the total number of clusters C, in which $\sum_{t=1}^{C} T_c = T$ is the total number of types to be clustered:

$$f(\mathbf{d}) = \Pr(T_1, \ldots, T_c \mid C) \Pr(C). \tag{3.4}$$

We assume a uniform discrete prior for the total number of clusters,

$$\Pr(C = c) = 1/T, \quad c = 1, \ldots, T, \tag{3.5}$$

and the uniform multinomial-Dirichlet distribution of Heard *et al.* (2006) for the total number of observations given the number of clusters, yielding

$$f(\mathbf{d}) = \Pr(T_1 \ldots, T_c, C) \propto \frac{(C-1)! T_1! \ldots T_C!}{(T+C-1)!}. \tag{3.6}$$

This prior favours small numbers of clusters and can be easily evaluated. The prior probabilities for $T = 3$ are computed in Table 3.1.

The prior (3.6) allows empty clusters, which is not an issue in hierarchical clustering, because dropping an empty cluster always makes the partition more probable. Another nice property of the assumed prior is that having all data in one cluster has the largest probability. This is useful when the joint density is almost equal for different partitions, because the prior forces the posterior to have a small number of clusters. In other words a mixture with more components is chosen only when it is really necessary.

The assumed prior can be perturbed in many ways, by assigning a non-uniform prior for the number of clusters (3.5), or by putting another distribution for the number of observations in the clusters, (3.6).

Different priors have been proposed in the literature. For example Tadesse *et al.* (2005) propose a truncated Poisson distribution for the number of clusters. Crowley (1995) and McCullagh and Yang (2006) propose another prior for the number of observations in clusters, and Booth *et al.* (2008) argue that this prior has the desirable property of being consistent, in the sense that groupings have the correct marginalization properties. One may simply assume a uniform prior for all partitions, that is considering any partitioning equally likely. This is not so appropriate, because when $C = 1$ there is just one way of constructing a nonempty partition, but when $C = 2$ there are $2^T - 1$ ways. Therefore, the posterior under this prior often yields a grouping with a lot of clusters.

3.2 Hierarchical Bayesian Clustering

Suppose y_{vctr} is the rth replicate of type t in cluster c measured on variable v, and y with fewer indices refers to an appropriate vector of data. This is the same notation as in Chapter 2 except c that there represented the class, here refers to the cluster. Assume that grouping is shown by a label vector $\mathbf{d}$, uniquely labelled as described in Section 3.1.1. In order to implement hierarchical clustering with the Bayesian models proposed in Chapter 2, evaluation of the marginal posterior for any data configuration represented by vector $\mathbf{d}$ is required.

The marginal posterior of the clustering may be written as

$$f(\mathbf{d} \mid y) = k^{-1} f(y \mid \mathbf{d}) f(\mathbf{d}). \tag{3.7}$$

We may ignore $k > 0$ in calculations because for a fixed number of types T, k is a constant and hence plays no role in inference and analysis. Hence, according to (3.7) for evaluation of the marginal posterior, it is only required to evaluate the prior and the marginal density. The prior $f(\mathbf{d})$ is defined in (3.4) and we assume it to be product of a uniform discrete distribution with a uniform multinomial-Dirichlet distribution. The joint density for data with $C = \max(\mathbf{d})$ clusters can be evaluated as

$$f(y \mid \mathbf{d}) = \prod_{c=1}^{C} f(y_c) = \prod_{v=1}^{V} \prod_{c=1}^{C} f(y_{vc}). \tag{3.8}$$

We just consider our proposed variable selection models, since models without variable selection can be obtained by setting the hyper-parameter $q = 1$. For the variable selection models, $f(y_{vc})$ as calculated in Section 2.5 takes the form

$$f(y_{vc}) = q \left\{ p f_1(y_{vc}) + (1-p) \prod_{t=1}^{T_c} f_0(y_{vct}) \right\} + (1-q) \prod_{t=1}^{T_c} f_0(y_{vct}), \tag{3.9}$$

where f_1 differs for the Gaussian and the asymmetric Laplace models, but f_0 is the same for both models. The density $f(y_{vc})$ has desirable properties as follows. For $q = 0$ or $p = 0$ the data density becomes

$$f(y \mid \mathbf{d}) = \prod_{v=1}^{V} \prod_{c=1}^{C} \prod_{t=1}^{T_c} f_0(y_{vct}) \tag{3.10}$$

which is product of the density f_0 of all types and hence is independent of the data configuration $\mathbf{d}$. Thus when $p = 0$ or $q = 0$, the joint densities of different configurations are identical and consequently the posterior of any configuration reduces to the prior. The posterior differs from the prior when $p > 0$ and $q > 0$. Therefore, p or q may be regarded as tuning parameters, used to flatten the posterior and jump from local modes when a stochastic search algorithm is implemented to sample from the posterior distribution.

The ordinary Gibbs sampler is known to be inefficient when searching over the space of partitions because it can get trapped in local modes, and split-merge algorithms have been proposed instead (Jain and Neal, 2004). The ordinary Gibbs sampler works as follows. An element of $\mathbf{d}$ is chosen at random, say d_t. Suppose in the current iteration the number of clusters is C and the sampled individual belongs to cluster c ($d_t = c$). In order to go to the next Gibbs sampling iteration all possible moves are proposed for that element, $d_t = \{1, \ldots, C_t\}$, in which C_t equals the current number of clusters C if cluster c is a singleton, and equals $C + 1$ otherwise (Chen *et al.*, 2006). The split-merge algorithms are similar unless additional moves are considered, like choosing two clusters at random and merging them, or choosing a cluster at random and randomly splitting it into two (Booth *et al.*, 2008). The additional moves are necessary to jump from the local modes because they provide access to partitions which are likely to have large posterior probabilities but are unlikely to be proposed by Gibbs sampling. An alternative to the split-merge approach is to flatten the posterior distribution that is letting the Gibbs sampler move more freely and gradually tempering the sampler as the Markov chain moves. This is called *reverse annealing* (Medvedovic, 2000; Medvedovic *et al.*, 2004) and can be easily implemented in our proposed models by fixing q and treating p as the annealing parameter. We demonstrate the usefulness of this technique using a toy example.

We construct toy data as follows. Consider bivariate data with twenty observations, and all data points sampled from univariate standard Gaussian distribution. This creates a dataset from one cluster. In order to see the effect of the reverse annealing, assume a uniform discrete distribution as the clustering prior, $f(\mathbf{d})$. At each Gibbs sampler step there are C or $C+1$ moves with a probability associated to each. There is a move with a high transition probability if the range $(\max - \min)$ of the log posterior probabilities is large. If this range is zero all moves are equally likely. Figure 3.4 shows the range of the log probabilities for Gibbs sampler of the toy data for the Gaussian variable selection model with $\sigma^2 = 1$, $\sigma_\eta^2 = 0$, $\sigma_\theta^2 = 1$, $\mu = 0$, $q = 1$ and different values of p, confirming that the range approaches zero, that is the chain tends to propose equally likely moves, when p is decreased.

Closed form marginal posteriors allow fast calculation of the probability

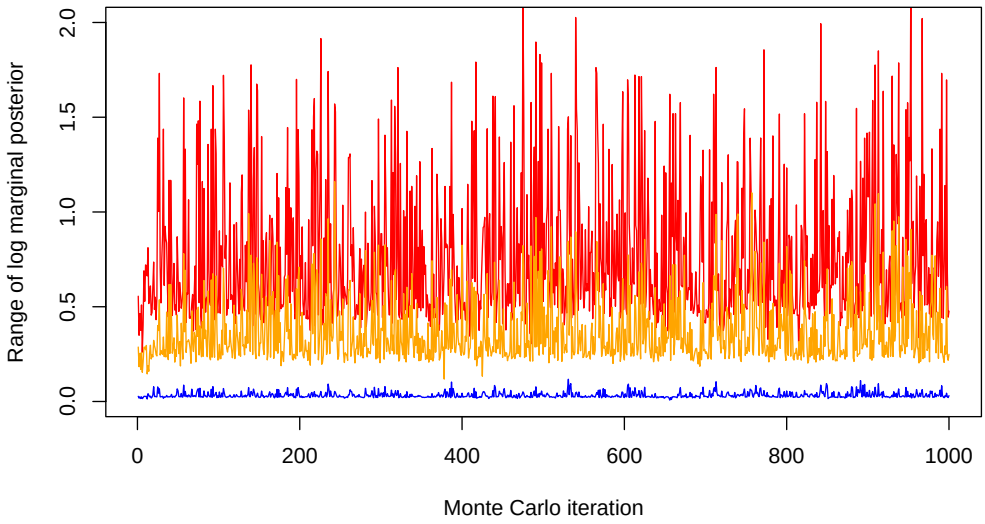

Figure 3.4: Log marginal posterior range of ordinary Gibbs sampler with 1000 iterations for the Gaussian model, model parameters are chosen as $\sigma^2 = 1$, $\sigma_\eta^2 = 0$, $\sigma_\theta^2 = 1$, $\mu = 0$, $q = 1$, and $p = 0.95$ (red), $p = 0.5$ (orange) and $p = 0.05$ (blue).

of any data configuration. This is very useful to construct a dendrogram, which is not easily feasible with other models. Practitioners like to see dendrograms because they give visual guides of how groupings may change if one chooses different numbers of clusters. Distance-based dendrograms give no guide of where to cut the tree, but using a model-based dendrogram allows us to provide a criterion: we cut the tree where the marginal posterior is maximised.

The dendrogram using the log posterior as the similarity measure provides a probabilistic interpretation of the tree.

In the following we describe how agglomerative clustering can be implemented using our suggested models.

We start with each observation as a single cluster, that is the uniquely labelled vector $\mathbf{d}$, is an increasing integer vector with elements all different, starting from 1 and ending with T. Hence the number of clusters is $C = T$

and the number of types in cluster c is $T_c = 1$, for all $c = 1, \ldots, C$. In the first step, all pairwise merges are considered and $\mathbf{d}$ is updated accordingly. For each pairwise merge, the marginal posterior (3.7) is calculated, the merge that maximises (3.7) is applied, and $\mathbf{d}$ is updated for the new grouping. We keep $g_c = \log f(\mathbf{d} \mid y)$, the log marginal posterior for the best merge having c clusters to use as the dendrogram height. Assume that the best merge according to (3.7) proposes to join cluster c_1 and cluster c_2. Thus, for the types that are joined, $\mathbf{d}$ get the same integer labels and for the new merged cluster c, $T_c = T_{c_1} + T_{c_2}$. The algorithm then considers all pairwise merges, and continues until all clusters are merged and all types are in one cluster.

The best grouping found using the posterior as the objective function on the ordered path found by the agglomerative method is the one that maximises g_c across $c = 1, \ldots, T$. It is clear from the agglomerative clustering procedure that the groupings associated to g_c are sorted in agglomerative order with increasing c, so a dendrogram representation is possible. In order to draw a dendrogram a monotone height is required, but g_c is not necessarily monotone and we use the following transformation. Suppose $g_{\max} = \max(g_c)$, and $c_{\max} = \text{argmax}(g_c)$ is the number of clusters that maximises g_c. For $c < c_{\max}$ we define the height of the dendrogram $h_c = g_c - g_{\max}$, which is negative, and for $c > c_{\max}$, $h_c = g_{\max} - g_c$, which is positive. By definition, h_c is monotone if g_c is unimodal, which is usually the case, and cutting the dendrogram at zero height gives the grouping that maximises g_c. If g_c is not unimodal the height is not monotone.

The Bayesian agglomerative clustering needs the evaluation of the marginal posterior and this is only possible after fixing the model parameters. Proper estimation of the parameters requires the true data grouping which is unknown, so we propose to estimate the model parameters at the first stage, that is considering every type as a single cluster and keep them fixed during agglomerative clustering.

Our experience is that the early stages of agglomerative clustering are important and parameter values play a crucial role. After joining some observations in a cluster, the parameters become less important and the marginal posterior is guided more by the grouped observations. One may re-estimate parameters after getting a reasonable grouping. However, when observations are merged to form a cluster, a smaller number of variable-cluster combinations is available compared with the first stage of clustering, and consequently estimation becomes much more difficult. Another approach is to assign a prior distribution to the model parameters, but this may disturb analytical tractability of the resulting marginal posterior and slows fitting the model.

Our hierarchical clustering method differs from previous researches in this field in various ways. It is different from Friedman and Meulman (2004) because we adopted a model and they define a distance. It is different from Hoff (2006), Heller and Ghahramani (2005) and Kim *et al.* (2006) who use Dirichlet process models. Our method is more similar to Tadesse *et al.* (2005) who propose finite Gaussian mixtures. Their variable selection needed reversible jump Markov chain Monte Carlo, but here we apply variable selection with closed form marginals, and hence trans-dimensional Markov chain Monte Carlo is unnecessary. Our algorithm is also close to Raftery and Dean (2006); we use the exact Bayes factor but they propose using an approximate Bayes factor with no dendrogram representation. Having an analytically tractable form marginal posteriors helps to give a fast clustering algorithm, and one can construct dendrogram trees with a probabilistic interpretation as in Heard *et al.* (2006). However, in order to get a closed form marginal posterior we must assume a model that imposes independent a posteriori variables. This seems restrictive, but the independence assumption for variables considerably facilitates fitting and does not affect the clustering performance a lot for high-dimensional data, as confirmed in the simulations of Chapter 4. Our approach is different from Heard *et al.* (2006) in two ways. First, we did not consider a prior for σ^2. Second, our model provides the possibility of selecting variable-cluster combinations through the Bernoulli variable γ_{vc} and selecting variables through δ_v. If $\delta_v = \gamma_{vc} = 1$, for $c = 1, \ldots, C, v = 1, \ldots, V$, and supposing $\sigma_\eta^2 = 0$, then our model reduces to that of Heard *et al.* (2006) with a degenerate prior for σ^2.

3.3 Computational Issues

3.3.1 General

The main difficulty of agglomerative clustering is fast evaluation of the data joint density $f(y \mid \mathbf{d})$. When the number of clusters is C, $C(C-1)/2$ merges are considered and because C varies from 1 to T, the total number of evaluations is $\sum_{C=1}^{T} C(C-1)/2$ which is of order $O(T^3)$. However, because our models impose independent variables, $f(y \mid \mathbf{d})$ reduces to $\prod_{v=1}^{V} \prod_{c=1}^{C} f(y_{vc})$ and agglomerative clustering is of order $O(VT^3)$, linear in terms of the number of variables V. This is encouraging because we assume a data structure such that T is small but V is large, and hence the algorithm is fast in this situation. However, evaluation of $f(y \mid \mathbf{d})$ may be time-consuming for large V or T and computational acceleration is required. In the following we describe several tricks to compute $f(y \mid \mathbf{d})$ reliably and to accelerate the agglomerative clustering algorithm.

3.3.2 Joint Density Acceleration

In order to decide which cluster must be merged, we need to evaluate a density of the form $f(y \mid \mathbf{d}) = \prod_{c=1}^{C} f(y_c)$, and this is computationally expensive if C is large, in the early stages of agglomerative clustering. A simple trick to rapidly evaluate $f(y \mid \mathbf{d})$, is to benefit from a property of the agglomerative clustering. In agglomerative method only two clusters will be joined and hence the evaluation of the density of two clusters with the past values of $f(y_c)$ suffices for evaluation of $f(y \mid \mathbf{d})$. Every time that we evaluate $f(y \mid \mathbf{d})$, only the joint density of the merging clusters is calculated and $f(y \mid \mathbf{d})$ is reconstructed by multiplying the lacking components.

3.3.3 Individual Density Computation

For a given configuration $\mathbf{d}$, the individual density $f(y_c)$ for the Gaussian and the asymmetric Laplace model is $f(y_c) = \prod_{v=1}^{V} f(y_{vc})$, where

$$f(y_{vc}) = q \left\{ p f_1(y_{vc}) + (1-p) \prod_{t=1}^{T_c} f_0(y_{vct}) \right\} + (1-q) \prod_{t=1}^{T_c} f_0(y_{vct}). \quad (3.11)$$

The density is composed of products and therefore it is easier to be computed on the log scale. Suppose we denote

$$\begin{aligned} lq_1 &= \log\left\{ pf_1(y_{vc}) + (1-p)\prod_{t=1}^{T_c} f_0(y_{vct}) \right\}, \\ lq_0 = lp_0 &= \sum_{t=1}^{T_c} \log f_0(y_{vct}), \\ lp_1 &= \log f_1(y_{vc}), \end{aligned} \tag{3.12}$$

then

$$l = \log f(y_c) = \log\{q\exp(lq_1) + (1-q)\exp(lq_0)\}. \tag{3.13}$$

When lq_0 and lq_1 are both very small or very large, computation of l is troublesome, and computer memory may overflow or l may be evaluated as zero. In order to avoid this problem we evaluate l in (3.13) after factorising $\exp(lq_1)$ as

$$l = lq_1 + \log\left\{q + (1-q)\exp(lq_0 - lq_1)\right\}. \tag{3.14}$$

This is appropriate when $lq_1 > lq_0$, because the exponent function in (3.14) doesn't explode. For $lq_0 \geq lq_1$ it is more appropriate to evaluate an analogous expression, i.e. factorising $\exp(lq_0)$ in (3.13),

$$l = lq_0 + \log\left\{1 - q + q\exp(lq_1 - lq_0)\right\}. \tag{3.15}$$

The log density lq_0 is straightforward to calculate from (2.24), but lq_1 takes a similar form as in (3.13), i.e.,

$$lq_1 = \log\left\{p\exp(lp_1) + (1-p)\exp(lp_0)\right\}, \tag{3.16}$$

and the same problem is encountered, so a similar trick is applied.

3.3.4 Density of the Gaussian Model

As described in Section 3.3.3, evaluation of individual densities requires evaluation of lp_1 for each variable and according to computations of Section 2.5.4, $lp_1 = \log f_1(y_{vc})$ corresponds to logarithm of a multivariate Gaussian density

with mean $\mu\mathbf{1}$ and covariance matrix $\mathbf{\Sigma}$, in which $\mathbf{1}$ is a vector of ones of length $p = \sum_{t=1}^{T_c} R_{ct}$ and $\mathbf{\Sigma}$ is a $p \times p$ positive definite matrix. The diagonal elements of the covariance matrix $\mathbf{\Sigma}$ equal $\sigma^2 + \sigma_\eta^2 + \sigma_\theta^2$ and the off-diagonal elements equal $\sigma_\eta^2 + \sigma_\theta^2$ if the observations are replications of the same type, and equal σ_θ^2 otherwise.

A p-variate Gaussian density ϕ has the form

$$\log \phi(y_{vc}) = -\frac{p}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}(y_{vc} - \mu\mathbf{1})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(y_{vc} - \mu\mathbf{1}). \quad (3.17)$$

Hence, evaluation of the density requires computation of the Mahalanobis distance $(y_{vc} - \mu\mathbf{1})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(y_{vc} - \mu\mathbf{1})$ and the log determinant of $\mathbf{\Sigma}$. In order to efficiently compute these two, assume $\mathbf{B}_{p\times p}$, an upper-triangular matrix, is the Cholesky decomposition of $\mathbf{\Sigma}$, that is $\mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{\Sigma}$. The Cholesky decomposition of a positive definite matrix is efficiently implemented in `Fortran` and the code is available from the `Fortran-NAG` library. Because $\mathbf{B}$ is upper-triangular, a solution to the system of linear equations

$$\mathbf{B}\mathbf{x} = (y_{vc} - \mu\mathbf{1}) \quad (3.18)$$

can be easily obtained by back-solving. Hence, $\mathbf{x} = \mathbf{\Sigma}^{-\frac{1}{2}}(y_{vc} - \mu\mathbf{1})$ might be used to evaluate the Mahalanobis distance as

$$\mathbf{x}^{\mathrm{T}}\mathbf{x} = \sum_{i=1}^{p} x_i^2 = (y_{vc} - \mu\mathbf{1})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(y_{vc} - \mu\mathbf{1}), \quad (3.19)$$

in which x_i represents the ith element of the vector $\mathbf{x}$.

Once the Cholesky decomposition of $\mathbf{\Sigma}$ is computed, the eigenvalues λ_i are also available. Denoting the diagonal elements of $\mathbf{B}$, by b_{ii}, we have $b_{ii} = \lambda_i^{\frac{1}{2}}$, and hence

$$\log|\mathbf{\Sigma}| = \sum_{i=1}^{p}\log\lambda_i = 2\sum_{i=1}^{p}\log b_{ii}. \quad (3.20)$$

The log density can be obtained by replacing the Mahalanobis distance (3.19) and the log determinant (3.20) in (3.17), yielding

$$\log \phi(y_{vc}) = -\frac{p}{2}\log 2\pi - \sum_{i=1}^{p}\log b_{ii} - \frac{1}{2}\sum_{i=1}^{p} x_i^2.$$

The required log density l can be obtained by putting pieces together and replacing them in (3.16), and then in (3.13). We need to apply this procedure for all vectors of data y_{vc}, $(v = 1, \ldots, V,\ c = 1, \ldots, C)$. We can save computational time for data in the same cluster but another variable, say $y_{v'c}(v' \neq v)$, because for $y_{v'c}$, the covariance matrix $\mathbf{\Sigma}$ and hence $\mathbf{B}$ is unchanged. Therefore, we do not need to re-calculate the Cholesky decomposition of $\mathbf{\Sigma}$. However, the back-solving must be updated according to the new data in $\mathbf{Bx} = y_{v'c} - \mu\mathbf{1}$, and the Mahalanobis distance must be recomputed using the new $\mathbf{x}$.

3.3.5 Density of the Asymmetric Laplace Model

The density $lp_1 = \log f_1(y_{vc})$ takes a complicated form as shown in (2.28). However, the computational difficulty arises only in calculation of

$$|\mathbf{A}|,\ \mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L,\ \mathbf{b}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R,\ \mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L,\ \mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R,\ \mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_L,\ \mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_R, \tag{3.21}$$

and Φ, the standard Gaussian cumulative distribution function. The cumulative Gaussian distribution function is already available in the `Rmath-C` library and evaluation of the quantities in (3.21) is similar to the Gaussian case explained in Section 3.3.4. First we calculate the upper-triangular Cholesky decomposition of $\mathbf{A}_{p\times p}$, say $\mathbf{B}_{p\times p}$, in which $p = T_c$. Hence

$$\log|\mathbf{A}| = 2\sum_{i=1}^{p}\log b_{ii},$$

and then by back-solving the following systems of linear equations we find vectors $\mathbf{x}_{\mathbf{b}_L}, \mathbf{x}_{\mathbf{b}_R}, \mathbf{x}_{\mathbf{d}_L}, \mathbf{x}_{\mathbf{d}_R}$,

$$\mathbf{B}\mathbf{x}_{\mathbf{b}_L} = \mathbf{b}_L,\ \mathbf{B}\mathbf{x}_{\mathbf{d}_L} = \mathbf{d}_L,\ \mathbf{B}\mathbf{x}_{\mathbf{b}_R} = \mathbf{b}_R,\ \mathbf{B}\mathbf{x}_{\mathbf{d}_R} = \mathbf{d}_R.$$

Therefore, the required quantities are

$$\begin{aligned}\mathbf{b}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L &= \mathbf{x}_{\mathbf{b}_L}^{\mathrm{T}}\mathbf{x}_{\mathbf{b}_L},\ \mathbf{b}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R = \mathbf{x}_{\mathbf{b}_R}^{\mathrm{T}}\mathbf{x}_{\mathbf{b}_R},\ \mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_L = \mathbf{x}_{\mathbf{d}_L}^{\mathrm{T}}\mathbf{x}_{\mathbf{b}_L},\\ \mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}_R &= \mathbf{x}_{\mathbf{d}_R}^{\mathrm{T}}\mathbf{x}_{\mathbf{b}_R},\ \mathbf{d}_L^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_L = \mathbf{x}_{\mathbf{d}_L}^{\mathrm{T}}\mathbf{x}_{\mathbf{d}_L},\ \mathbf{d}_R^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{d}_R = \mathbf{x}_{\mathbf{d}_R}^{\mathrm{T}}\mathbf{x}_{\mathbf{d}_R}.\end{aligned} \tag{3.22}$$

	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
Gaussian	$\underset{(0.020)}{-6.622}$	$\underset{(0.006)}{0.977}$	$\underset{(0.008)}{3.187}$
Asymmetric Laplace	$\underset{(0.008)}{-6.227}$	$\underset{(0.003)}{0.976}$	$\underset{(0.003)}{2.96}$

Table 3.2: Least squares estimate of $\log_{10}\text{time} = \beta_0 + \beta_1 \log_{10} V + \beta_2 \log_{10} T$ and their standard errors, for the Gaussian and the asymmetric Laplace variable selection models.

The density $f_1(y_{vc})$ is obtained by putting the pieces together and replacing them in (2.28). For data in the same cluster but another variable, say $y_{v'c}$, the quantities $\mathbf{A}, \mathbf{d}_L$, and $\mathbf{d}_R$ are unchanged. Hence, we just need to update $\mathbf{x}_{\mathbf{b}_L}$ and $\mathbf{x}_{\mathbf{b}_R}$, replace them in (3.22) and evaluate $f_1(y_{v'c})$ with less computational effort.

3.3.6 Code Analysis

As we described in Section 3.3.1, apart from the optimisation required for parameter estimation, our clustering algorithm is of complexity order $O(VT^3)$. In this section we analyse our code, implemented in `C` with the help of the `Fortran-NAG` and `Rmath-C` libraries, and run from `R` (Chaudhary, 2007).

In order to analyse our computer code, a simple factorial experiment was performed with the number of variables $V = 50, 100, 200, 300, 500, 1000$ and the number of individuals $T = 10, 20, 30, 40, 50, 100, 200, 300$. The experiment is run on a desktop PC with Intel Core Duo processor 1.8 MHz, 1 GB RAM and Linux UBUNTU operating system. Each design is fitted 5 times using the Gaussian and the asymmetric Laplace models with variable selection and the time required for agglomerative clustering is saved in seconds.

The contour plots of $\log_{10}$ time required for clustering, using the Gaussian model and the asymmetric Laplace models are shown in Figure 3.5. The contour plots suggest a linear regression of $\log_{10}$ time on $\log_{10} V$ and $\log_{10} T$. Therefore, the regression model $\log_{10}\text{time} = \beta_0 + \beta_1 \log_{10} V + \beta_2 \log_{10} T$ is fitted and coefficients $(\beta_0, \beta_1, \beta_2)$ are estimated using least squares. The estimated coefficients and their standard errors are shown in Table 3.2.

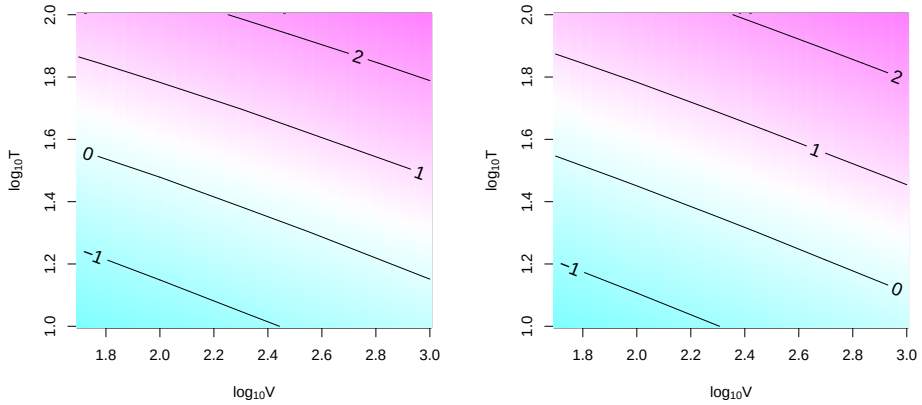

Figure 3.5: Contour plot of $\log_{10} V$ versus $\log_{10} T$ evaluated on $\log_{10}$ of time required for clustering in seconds with T individuals and V variables using the Gaussian variable selection model (left panel), and the asymmetric Laplace variable selection model (right panel). Colour is used for better visualisation.

According to Table 3.2 the regression parameter β_1 is estimated close to 1 and β_2 close 3 for the Gaussian and the asymmetric Laplace models. This is what we expected, because our agglomerative clustering is of order $O(VT^3)$ and hence $\log_{10}$ time regressed on $\log_{10} V$ should give a coefficient close to one and regressed on $\log_{10} T$ close to three. However, agglomerative clustering for the asymmetric Laplace model is implemented more efficiently than for the Gaussian model for large T, because β_2 for the asymmetric Laplace model is significantly smaller than β_2 for the Gaussian model at 95% level.

The fitted linear model in Table 3.2 can be used to predict the time required for agglomerative clustering for large T or V. However, we note that β_0 is often computer-dependent, and may change when the same algorithm is applied on another machine. The time needed for clustering $T = 100$ types measured on $V = 5000$ variables is about 39 minutes for the Gaussian model and 33 minutes for the asymmetric Laplace model.

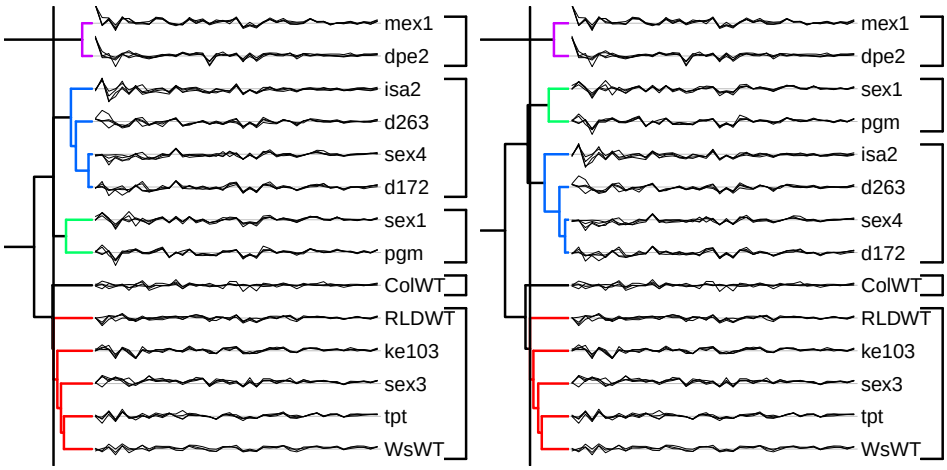

Figure 3.6: Clustering of the metabolite data using the Gaussian model (left panel) and the asymmetric Laplace model (right panel). The dendrogram is shown at the left of the profile plots, and the optimal clustering is shown by the vertical line cutting the dendrogram. The optimal grouping is represented at the right of profiles.

3.4 Examples

3.4.1 Metabolite Data

In the metabolite data of Section 1.3.1 the genetic background for some of the mutants is known and for some of them it is unknown. Biologists are interested to know which one of the known and unknown mutants, with probably different genetic backgrounds, show similar metabolite patterns. Answering this question led us to provide a clustering procedure equivalent to the classification described in Chapter 2.

The first step of clustering with the proposed models is estimation of the model parameters. Considering each type as a separate cluster, the likelihood (2.3) is maximised. This yields the estimates reported in Table 2.1, on page 49. Then agglomerative clustering using the Gaussian and the asymmetric Laplace models is implemented and the model parameters are kept fixed during construction of the tree. The resulting dendrograms are shown in Figure 3.6.

The dendrogram built using the Gaussian model and using the asymmetric Laplace model are not identical, but they are very similar. Both methods propose five groups. Mutants *mex1* and *dpe2* merge together and clearly are distinguished as a separate cluster, because their cluster is the last group to join the other types. This is coherent with their metabolite profile in Figure 1.2, because *mex1* and *dpe2* are mutants having an extreme jump on metabolite *maltose.MX1* and clearly distinguishable from the other plants. Another cluster consists of *isa2*, *d263*, *sex4* and *d172*, being the closest group to *sex1* and *pgm* which merge their own cluster. The wild types *RLDWT* and *WsWT* merge together with *ke103*, *sex3*, and *tpt*; these are types with flat profiles. However, *ColWT*, another wild type having a flat profile, is detected to be a single cluster, but close to the other wild types.

The clustering of the Gaussian and the asymmetric Laplace models agrees broadly with the classification reported in Table 2.2. For example probabilities for the wild types *RLDWT* and *WsWT* were spread out, meaning these mutants are close to each other, and clustering proposes these types to be in the same cluster. Mutant *sex1* has a high probability to be *pgm* and they fall into the same cluster. The wild type *ColWT* has a non-zero probability to be a new type and agglomerative clustering declares *ColWT* to be a single cluster. The unknown types *d263* and *d172* are classified to *sex4* and form a cluster with it. All classified types have zero probability to be *dpe2* and *mex1*; these two plants are joined and clearly declared to be different from the other mutants.

In Figure 3.7 (left panel) the marginal posterior is plotted across the number of clusters proposed by the agglomerative method for the Gaussian and the asymmetric Laplace models. Both curves are maximised at $C = 5$. The marginal posterior for the Gaussian model with $C = 4$ is almost the same; the difference is 0.18 on the log scale, so the posterior probability of merging *ColWT* with the other wild types is about 0.45. For the asymmetric Laplace model maximisation at $C = 5$ is clear. The asymmetric Laplace posterior is always greater than the Gaussian posterior from the beginning of the clustering procedure, that is $C = 13$, so the asymmetric Laplace model fits better. This is not surprising because the asymmetric Laplace model is more flexible than the Gaussian model; it has six parameters where as the

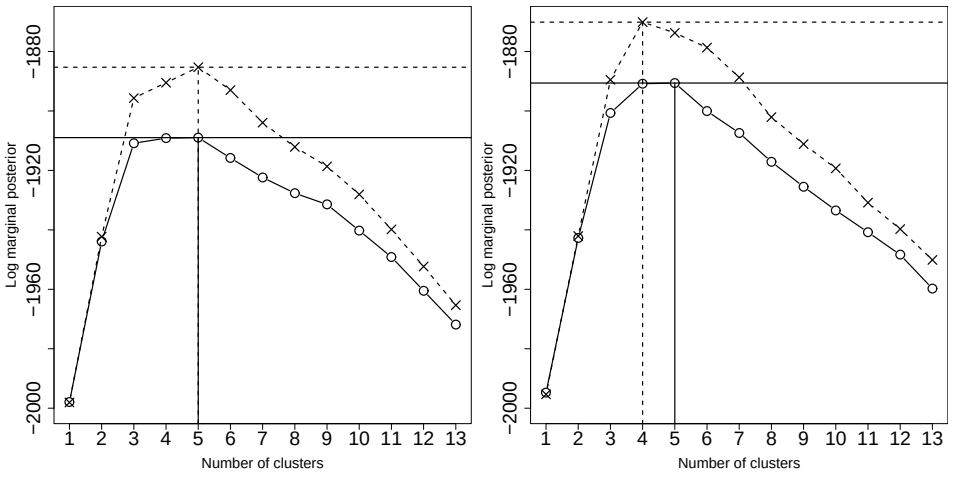

Figure 3.7: Log marginal posterior for the metabolite data as a function of the number of clusters proposed by the agglomerative method, for the Gaussian (circles) and the asymmetric Laplace (crosses) model. The left panel corresponds to the models without variable selection ($q = 1$) and the right panel corresponds to the variable selection models. The optimal numbers of clusters are shown by the vertical lines, and the values of the log marginal posterior at the optimum by the horizontal lines.

Gaussian model has five parameters.

The metabolite example has replicated measurements. In order to inspect the effect of replication we refit the agglomerative clustering using the Gaussian model, but considering each replicate as a single type. The model parameters are fixed the same as in the replicated case, and the resulting dendrogram is shown in Figure 3.8. This yields a dendrogram in which replicates of the same type are often located close to each other. The optimal grouping is not the same as the replicated case in Figure 3.6 (left panel), but is comparable. In the unreplicated clustering, replicates of *pgm, sex1, sex3, ke103, tpt, WsWT, RLDWT,* and *ColWT* are in one cluster, but in the replicated clustering they are grouped into three different clusters. Replications of *dpe2* are clustered in a separate group as well as *mex1*, but these two types are merged together in the replicated clustering.

In order to find out which metabolites are important, the variable selection extension of the Gaussian and the asymmetric Laplace models are fitted

and their dendrograms are represented in Figure 3.9. Like Section 2.4.1 in the parameter estimation step, just p and q are estimated and the other parameter values are fixed to the values already estimated from the Gaussian and asymmetric Laplace models with $q = 1$, giving $\hat{p} = 0.458$ and $\hat{q} = 0.156$ for the Gaussian and $\hat{p} = 0.83$ and $\hat{q} = 0.183$ for the asymmetric Laplace model. The Gaussian variable selection model proposes $C = 5$ clusters, the same as in Figure 3.6, but the asymmetric Laplace variable selection model clearly suggests $C = 4$, merging *ColWT* with the other wild types. The multidimensional scaling of the negative log marginal posterior as a distance for the first step of the agglomerative clustering is plotted for the Gaussian and the asymmetric Laplace variable selection in Figure 3.10, confirming that the distance based on the asymmetric Laplace variable selection model finds the wild type *ColWT* to be closer to the other flat profiles compared with the distance created by the Gaussian model. Hence it is not surprising to see that the optimal grouping with the Laplace model merges *ColWT* with the other wild types.

As explained in Section 2.3.2, the Bayes factor B_v^{10} can be regarded as a measure of importance of the variables that separate the data according to a given grouping. Hence, we compute the Bayes factor for the optimal grouping found by our agglomerative method and sort metabolites with respect to B_v^{10}. The sorted metabolites and the log Bayes factor values are shown in Figure 3.11. The six most important metabolites have the same ordering using the Gaussian and the asymmetric Laplace models. They are *maltose.MX*, *X18*, *raffinose2*, *X16*, *glumatic.3*, and *mannitol*; these have $\log B_v^{10} > 3$. The metabolites having positive log Bayes factors using the Gaussian model are also found to have positive log Bayes factor using the asymmetric Laplace model. However, there are metabolites with negligible positive log Bayes factors using the asymmetric Laplace model but having a negative log Bayes factor using the Gaussian model. According to both models, 11 metabolites have positive log Bayes factors.

In order to compare our results with an existing clustering algorithm that both clusters data and computes variable importance, we applied the COSA clustering approach of Friedman and Meulman (2004). This method does not allow repeated measurements. We ran the COSA twice, first on

the mean of the profiles to somehow incorporate the replication information, but this gave zero importance for all metabolites, for a subset consisting of mutants *mex1* and *dpe2* that we were interested in. In the second run we considered the data to be unreplicated. The COSA algorithm is a distance-based method that uses the weighted distance defined in (3.1) and updates the weights according to the contribution of each subset to the clustering. Thus the importance, the weight, can be computed only for a specified group of data.

The COSA dendrogram using the average linkage method is shown in Figure 3.12. Cutting the dendrogram at height 0.62 gives five groups. The first group (from left to right) involves the unknown mutants *d172*, *d263*, *isa2*, and *sex4*; the second involves the wild type *WsWT* and the mutants having flat profiles *ke103*, *tpt*, *sex4*, and *sex3*; the third consists of a replicate of *WsWT* mistakenly grouped with *dp2* and *mex1*; the fourth consists of *pgm* and *sex1*; and the fifth group is the remaining types including a replicate of *dpe2* which is mistakenly grouped with the wild types *ColWT*, *RLDWT*, *sex4* and *sex3*. The clustering result is somehow in agreement with our results, except for a replicate of *dpe2* which is clearly distinguished on metabolite *maltose.MX1* is wrongly grouped with the wild types having flat profiles. A replicate of the wild type *WsWT* is also wrongly grouped with replicates of *dpe2* and *mex1*.

The importance of variables using the COSA algorithm is defined differently from that of ours and can be calculated only for one group (Friedman and Meulman, 2004). Importance plots corresponding to the five groups obtained by cutting the dendrogram of Figure 3.12 at height 0.62 are given in Figure 3.13. We just show the 12 most important metabolites.

The second cluster from left to right, obtained by cutting the tree of Figure 3.12, corresponds to the mutants having flat profiles, so all variables must have about the same importance values. This is confirmed in the importance plot of Figure 3.13, the top middle panel. The same holds for the fifth cluster which consists of mutants with flat profiles and the wild types, see the bottom middle panel of Figure 3.13. However, the fourth group consists of replicates of *pgm* and *sex1* which are different from the other types on metabolites *X18*, *X16*, and *raffinose2*, see Figure 2.8, and none of them

are recognised to be important by the COSA algorithm, see the bottom left panel of Figure 3.13. The third cluster which is shown by a rectangle in Figure 3.12 consists of replicates of *mex1* and *dpe2* that are clearly distinguished from the other mutants on *maltose.MX1*, which is not in the list of important clustering metabolites using COSA either, see the top right panel of Figure 3.13. The most important metabolite for this group is *ribose.MX* which is inconsistent with our analysis because it has a negative Bayes factor using both Gaussian variable selection model and asymmetric Laplace variable selection model, see Figure 3.11 and image plots of Figure 3.9. This has led us to investigate the COSA algorithm further on the metabolite example. We deleted the replicate of the wild type *WsWT.4* from the subset represented by rectangle in Figure 3.12, and recalculated the importances using the COSA method. Then, *maltose.MX1* appeared as the second important metabolite, but still *ribose.MX* is found to be the most important metabolite which seems unrealistic because the metabolite *ribose.MX* is flat across all types, see profile plots of Figure 2.8. Therefore, we conclude that the importance calculated using the COSA is sensitive to each member of the group and may give results inconsistent with ours.

All clustering results discussed in this section are somehow in agreement with the exploratory analysis presented in Section 1.4.2. Our proposed approach can be regarded as a method that coherently quantifies the exploratory plots of Figure 1.6 using a hierarchical Bayesian model.

Apart from parameter estimation, which took less than half a second, the time needed for agglomerative clustering took about a fifth of a second for all models on a common desktop computer, except the unreplicated fit which took 2.5 seconds. Applying the COSA algorithm on the unreplicated metabolite data took about a tenth of a second.

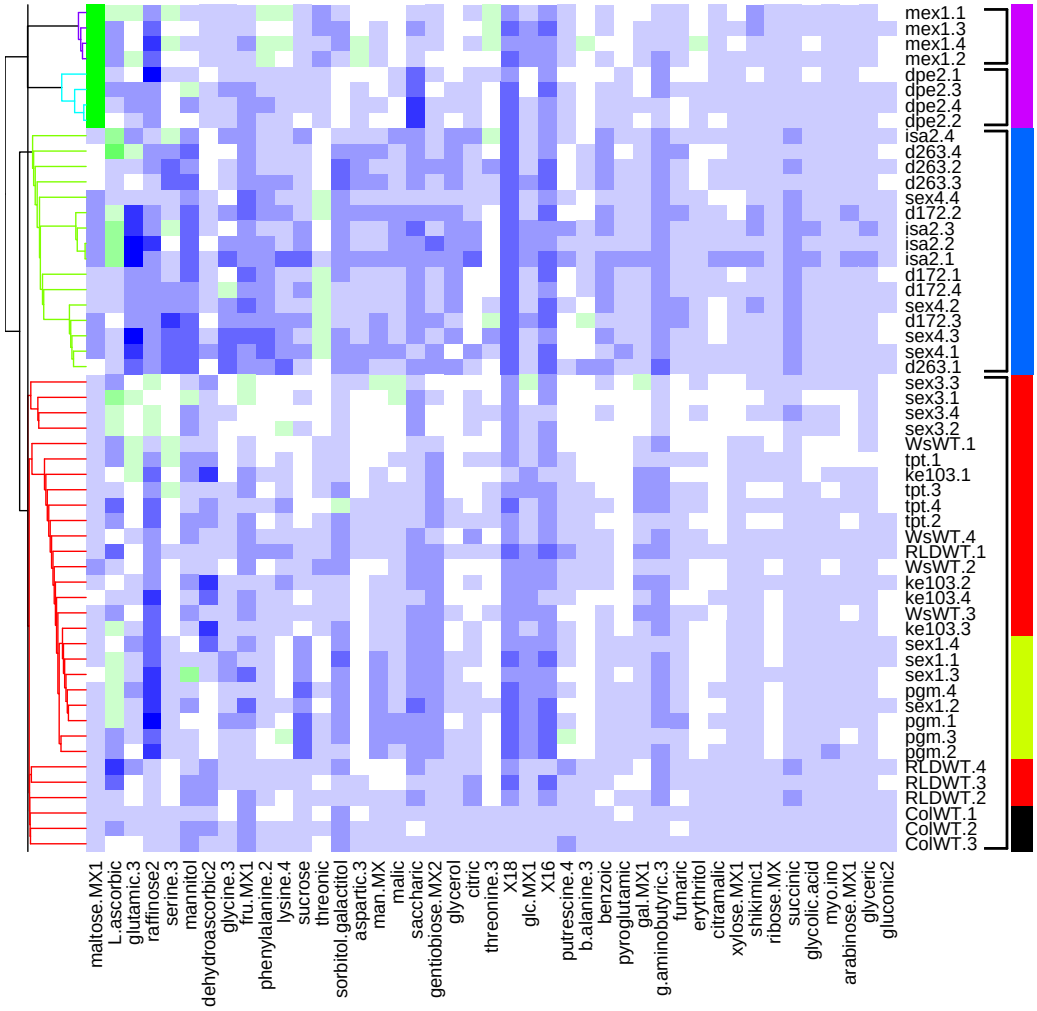

Figure 3.8: Dendrogram and optimal grouping found by the Gaussian model for the metabolite data, ignoring the replication information. The model parameters are the same as the replicated case. The vertical colour bar on the right refers to the optimal grouping found by the Gaussian model using the replication information. See also the left panel of Figure 3.6.

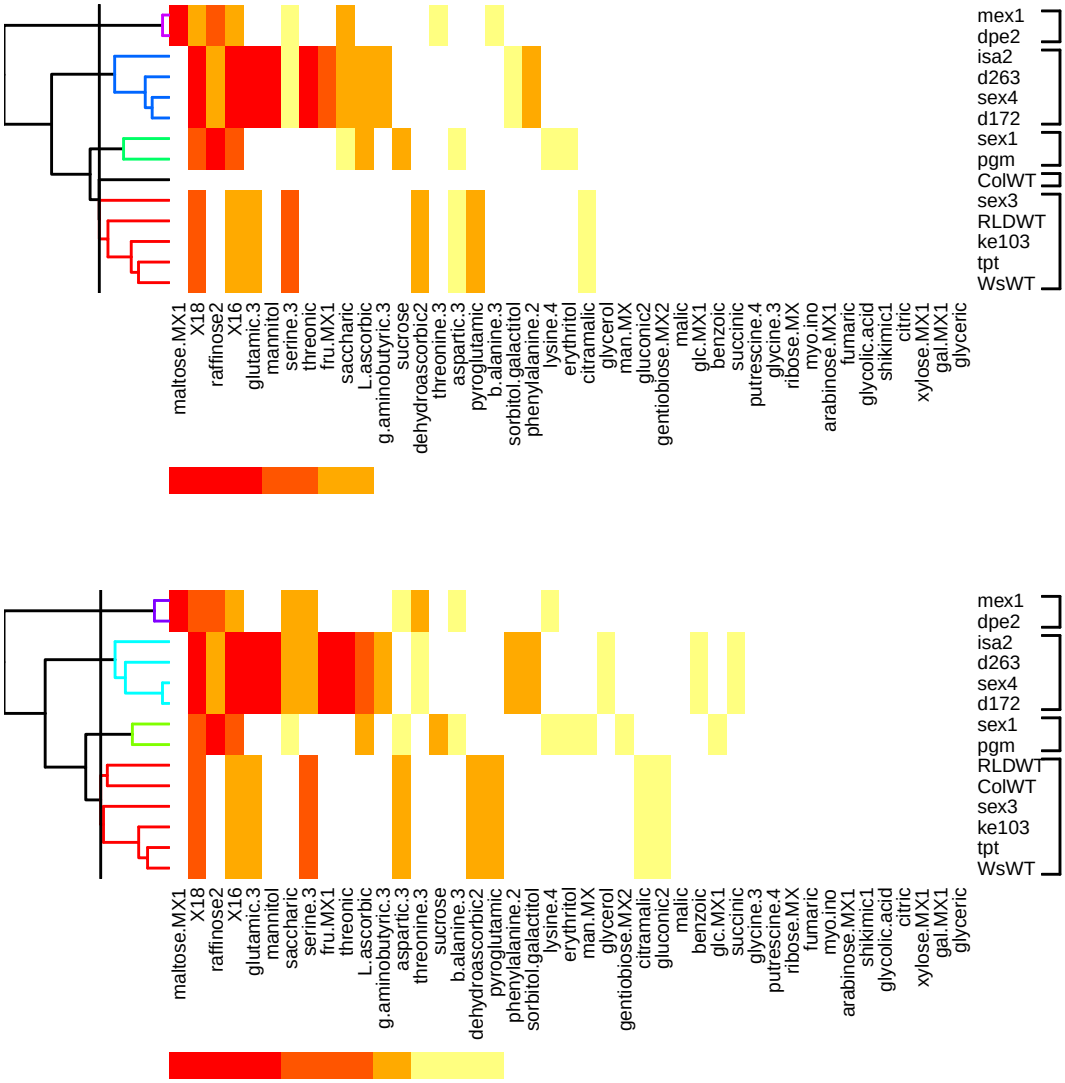

Figure 3.9: Clustering of the metabolite data using the Gaussian variable selection model (top panel) and the asymmetric Laplace variable selection model (bottom panel). The dendrogram obtained by the agglomerative method is shown at the left side of the image plot of $\log B^{10}_{vc}$, computed for the optimal grouping. Metabolites are sorted according to the Bayes factor B^{10}_{v}. The heat colours corresponds to the scale proposed by Kass and Raftery (1995), for more details see the caption to Figure 2.9. Bar plots of $\log B^{10}_{v}$ are shown in Figure 3.11.

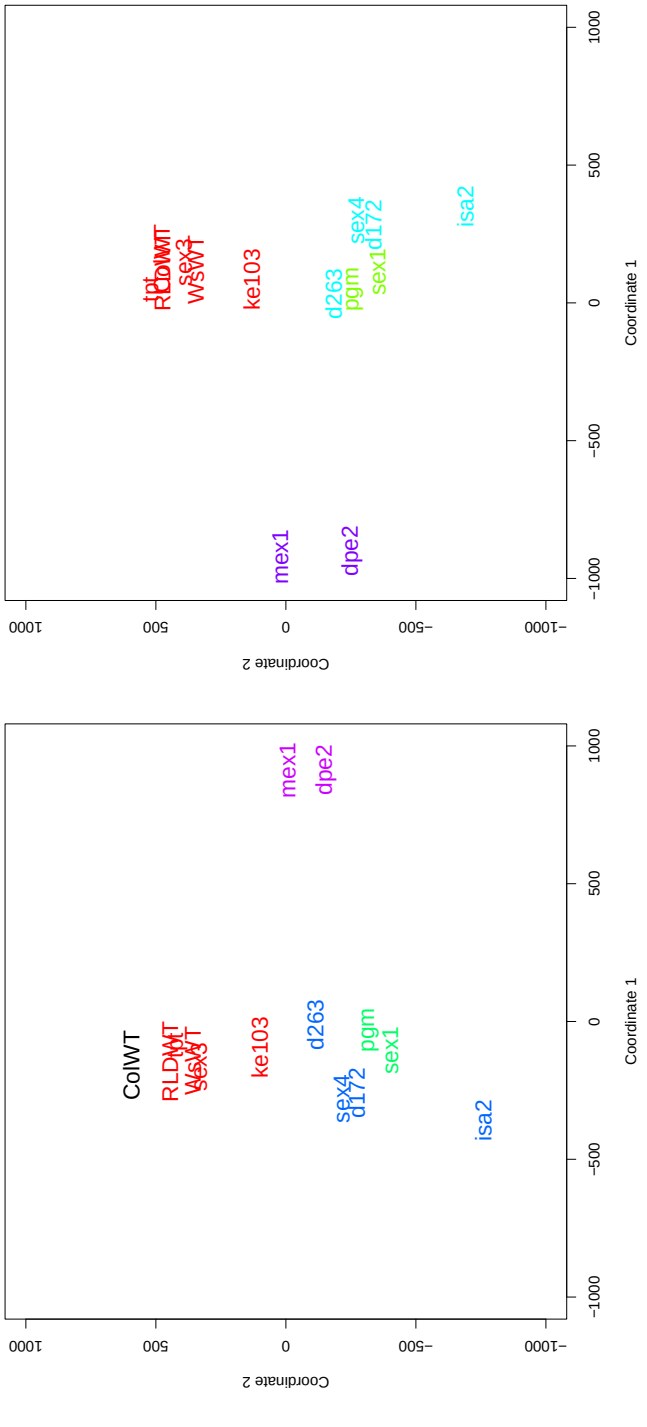

Figure 3.10: Multidimensional scaling of the negative log marginal posterior as a distance for the first step of agglomerative clustering using the Gaussian variable selection model (left panel) and using the asymmetric Laplace variable selection model (right panel). Colours correspond to dendrograms of Figure 3.9.

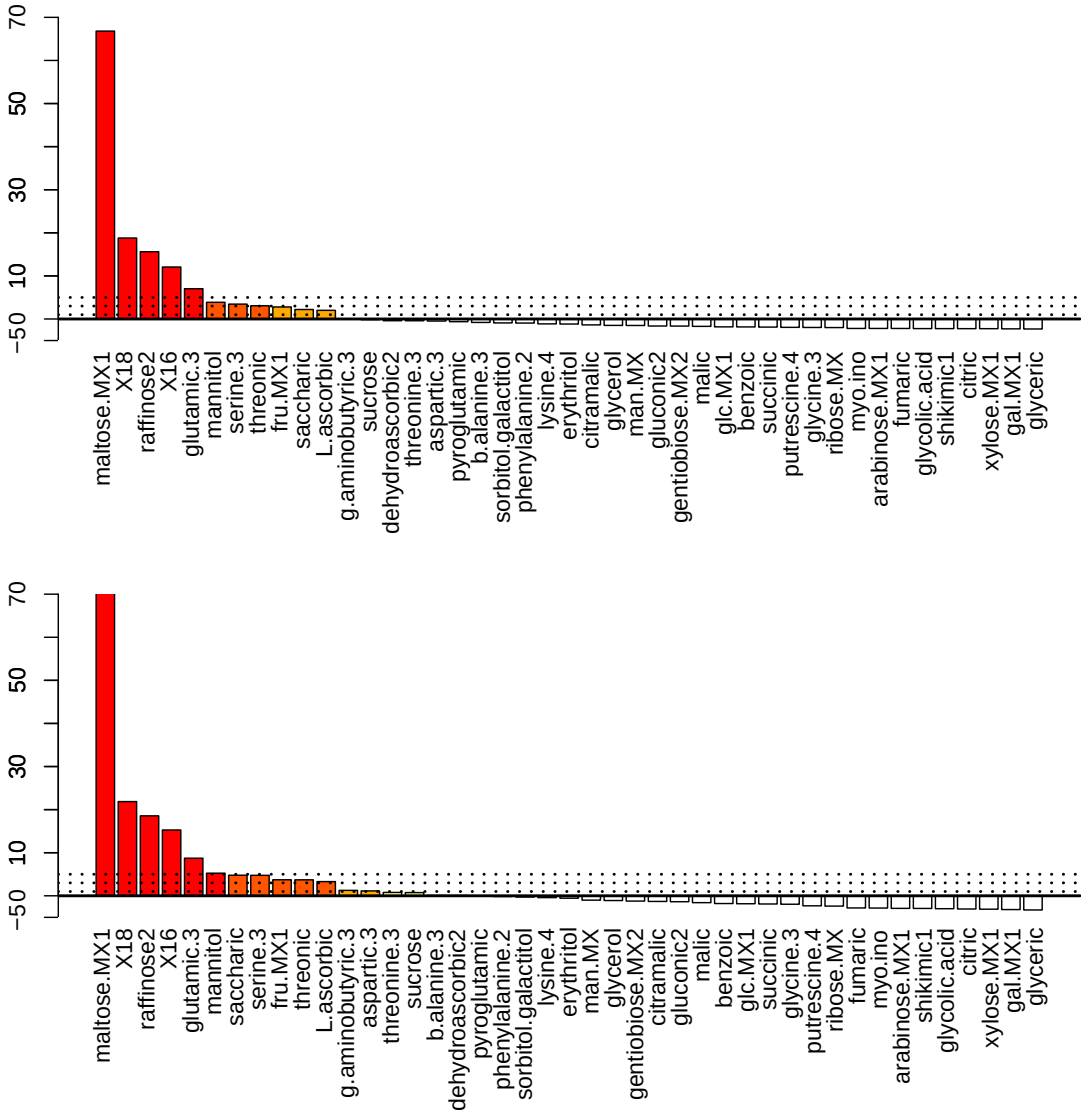

Figure 3.11: Log Bayes factor of variables, $\log B_v^{10}$, for the Gaussian (top panel) and the asymmetric Laplace (bottom panel) variable selection models. The Bayes factors are computed for the optimal grouping found by agglomerative clustering using the Gaussian model for the top panel and the asymmetric Laplace model for the bottom panel. The horizontal dotted lines represent the values used to categorise and colour the log Bayes factors. See also the caption to Figure 2.9. Image plots of $\log B_{vc}^{10}$ are shown in Figure 3.9.

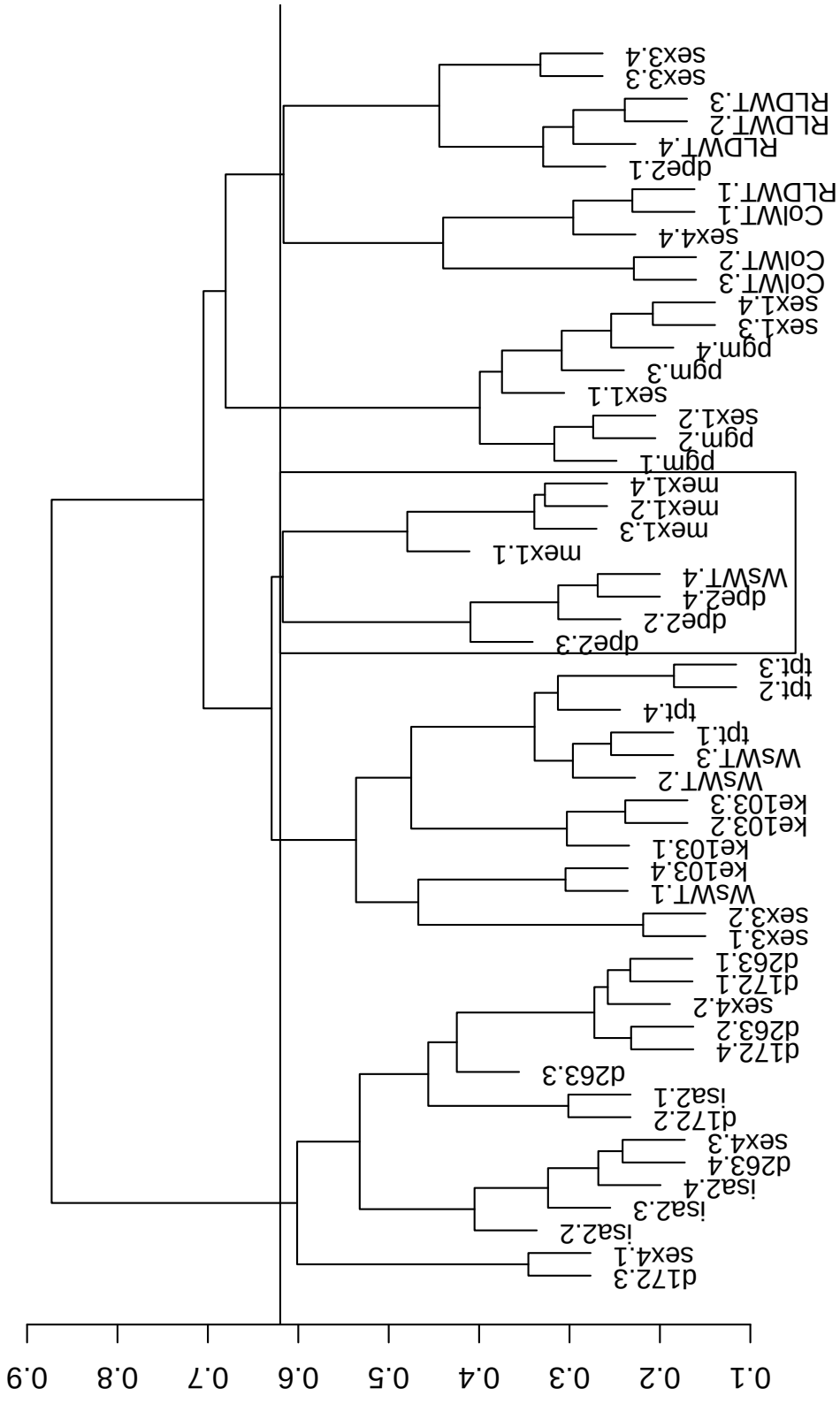

Figure 3.12: Dendrogram of the metabolite data obtained by COSA algorithm (Friedman and Meulman, 2004) using the average linkage method. We suggest cutting the dendrogram at height 0.62, which defines five clusters. A rectangle highlights the subset which we recalculated the importances after deleting the misclassified individual *WsWT.4*.

Figure 3.13: Importance plot of the COSA algorithm for the dendrogram of Figure 3.12 cut at height 0.62. The first cluster from left to right (top left), the second (top middle), the third highlighted by a rectangle (top right), the forth (bottom left), the fifth (bottom middle). The bottom right panel corresponds to the importance of the variables of the subset highlighted by a rectangle after deleting the misclassified subject *WsWT.4* from the subset.

3.4.2 Microarray Data

In this section we apply our clustering procedure to the microarray data of Section 1.3.2, which is an unreplicated dataset with a larger number of individuals (74 observations) and a higher number of dimensions (396 variables). We show the result of the Gaussian variable selection model, but fitting other models gives similar results. The model parameters are estimated treating every observation as a separate cluster. Since the data are unreplicated, the model is identifiable only with respect to $\sigma^2 + \sigma_\eta^2$. Therefore we estimate the parameters after fixing $\sigma_\eta^2 = 0$. The estimated parameters and their standard errors are $\sigma^2 = 0.547$ (0.007), $\sigma_\theta^2 = 1.486$ (0.052), $\mu = 0.013$ (0.005), $p = 0.716$ (0.021), $q = 0.393$ (0.026). The 95% confidence interval using profile likelihood for p is $(0.67, 0.76)$ and for q is $(0.34, 0.44)$. The profile likelihood confidence intervals for the other parameters are very close to the ones obtained using the standard errors reported above.

The data are divided into two groups using survival information of the patients. If gene pattern affects survival, clustering patients using their gene information must reflect the survival grouping. The data have been already analysed by Freije *et al.* (2004) and four groups were proposed in which 13 patients were misclassified. The agglomerative clustering dendrogram using the Gaussian variable selection model is shown in Figure 3.14, proposing $C = 10$ groups with 9 subjects misclassified. The number of genes having positive $\log B_v^{10}$ is 156 out of 396. This agrees with the estimated $q \approx 0.4$. The data are shown on the variables having positive log Bayes factors and on the misclassified individuals in Figure 3.15.

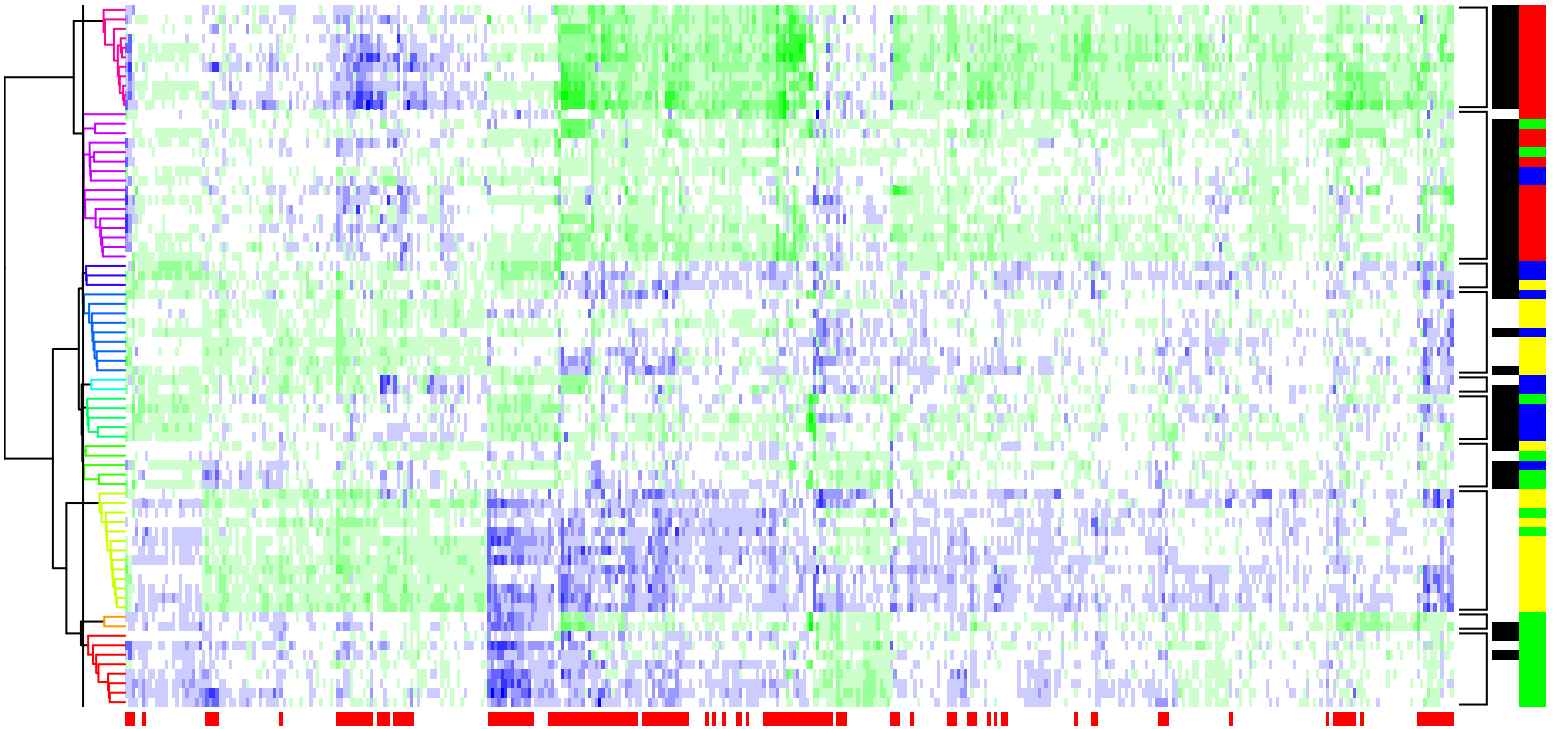

Figure 3.14: Microarray data clustering obtained using the Gaussian variable selection model with dendrogram at the left side of the image plot. On the right side of the image plot the optimal grouping is shown. The black and white vertical bar is white if the individual belongs to survival group 1 and is black if the individual belongs to survival group 2. The extreme right vertical bar corresponds to the four groups suggested by Freije *et al.* (2004). The horizontal heat bar at the bottom of the image plot refers to variables with positive $\log B_v^{10}$ computed for the optimal grouping; red is used if the log Bayes factor is positive.

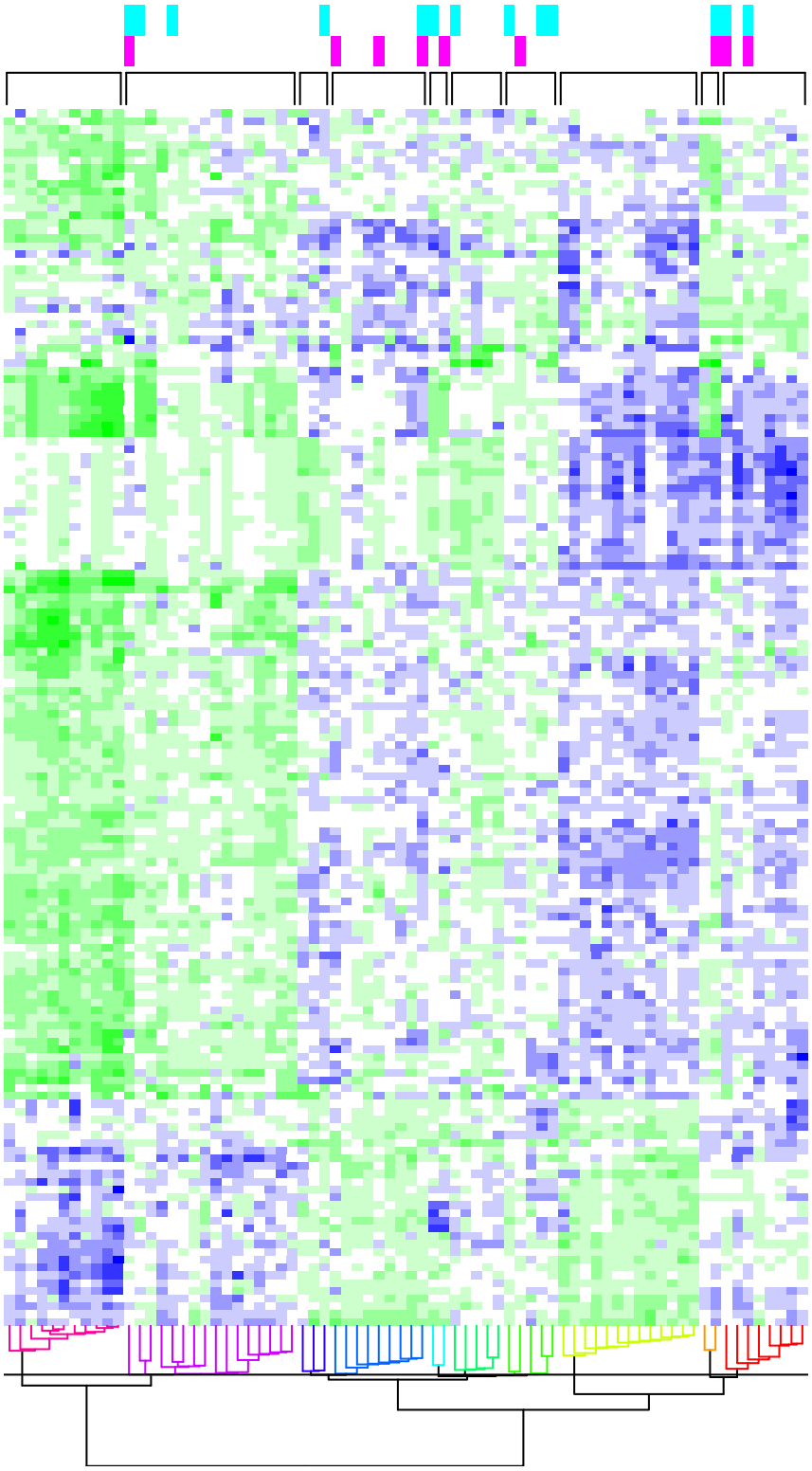

Figure 3.15: Dendrogram of the Gaussian variable selection model of the microarray data represented on the variables having positive $\log B_v^{10}$. Rectangles on the right show misclassified individuals: magenta is used to show misclassification of clustering with the Gaussian variable selection model, and cyan is used to show misclassification with clustering method proposed by Freije *et al.* (2004). See also Figure 3.14.

3.4.3 Image Data

In this section we present clustering with the Gaussian variable selection model on image data. Here our model assumptions are violated and the data are extremely high-dimensional. We took three grayscale portraits of R. A. Fisher, all of size 75×95 pixels, photos A, B, and C. Each pixel of a grayscale photo takes a value in the interval $[0, 1]$; zero if the pixel is pure black, one if is pure white and a value between zero and one otherwise. Each pixel is coded as a byte (eight bits), hence 256 gray levels are considered. We may rearrange the pixels of a portrait in a vector of dimension 7125, and consider each photo is a true profile with 7125 dimensions. Since true profile values are between zero and one, the Gaussian assumption for the true profiles is clearly wrong and since an image has an spatial pattern the independence assumption does not hold either. In order to violate the model assumptions further we create noisy profiles by adding to each pixel noise taken from continuous uniform distribution defined on interval $[0, 1]$. We make five observations of image A, ten of image B, and twenty of image C. Hence, these artificial data contain three unbalanced clusters. The portraits and a sample noisy observation of each image are shown in Figure 3.16.

In order to use our clustering method, first the median of each variable is subtracted and then our Gaussian variable selection model is fitted. The model parameters are estimated considering the observations to be unreplicated, so parameter estimation requires fixing $\sigma_\eta^2 = 0$. In optimisation of the likelihood function we encountered divergence of the optimisation routine, which is not surprising because the model distribution assumptions are badly wrong. In order to help the optimisation procedure, we set $\mu = 0$, a reasonable adjustment because after subtracting the median of each variable from the observations, μ approaches zero. The estimated parameters are $\hat{\sigma}^2 = 0.13$, $\hat{\sigma}_\theta^2 = 0.03$, $\hat{p} = 0.04$ and $\hat{q} = 0.54$. An agglomerative clustering dendrogram using the Gaussian variable selection model is shown in Figure 3.17, giving three groups in which all the noisy images are correctly grouped. Figure 3.18 shows $\log B_v^{10}$ referring to informative clustering pixels, and confirms that most of the pixels located around the face are useless. This is expected because photos are different mostly in the background of photos

which often have positive log Bayes factors.

The appropriateness of our clustering method as a device for image processing might be questioned in many ways. Our models give similar results for any rearrangement of the pixels, but photos and images have a strong spatial pattern which our models ignore. For applying our clustering method images should have the same size and this considerably restricts the application of our technique as an image processing tool. Above all, our clustering method is sensitive to translation, scaling and rotation of images which are minimal properties of a good image clustering procedure.

3.5 Discussion

This chapter demonstrated how a clustering method can be developed using a Bayesian hierarchical model. We used agglomerative clustering because the visual representation of grouping is possible when the marginal posterior is analytically tractable. This gives a dendrogram with a probabilistic interpretation. The method is automatic and can be run on both replicated and unreplicated data.

Computational issues related to such clustering methods were discussed and a computationally efficient technique for our proposed models was presented. Our codes are imported into `R` to benefit from its graphical facilities, but the code is written in `C` to gain computational speed. It will be released as an `R` package in the near future.

We showed the application of the method to metabolomic data, a microarray and an image example. Our clustering method works properly on various examples and is fast, especially for high-dimensional-low-sample-size situations.

We have assumed two different mixing distributions, a heavy tail and asymmetric distribution, and a Gaussian distribution. Apparently after proper estimation of the model parameters, the mixing distribution is not important in classification and clustering. This is confirmed in our examples and in our simulations. However, estimation of the model parameters may be difficult for some choices of the mixing distribution.

The main advantage of our clustering technique is to give variables an

importance measure that can be rapidly calculated and has a probabilistic interpretation. This is not easy to obtain using the existing clustering methods, as far as we know. An alternative is COSA (Friedman and Meulman, 2004), which is a distance-based method with no probabilistic interpretation.

A similar approach can be used to create a model-based clustering method analogue to a classification technique, for any Bayesian hierarchical model having closed form marginal posteriors.

Figure 3.16: Portraits A, B, and C of R. A. Fisher (top images) from left to right, all of size 75×95. Noisy photos (bottom images) are created by adding noise to each pixel, sampled independently from uniform distribution on the interval $[0, 1]$.

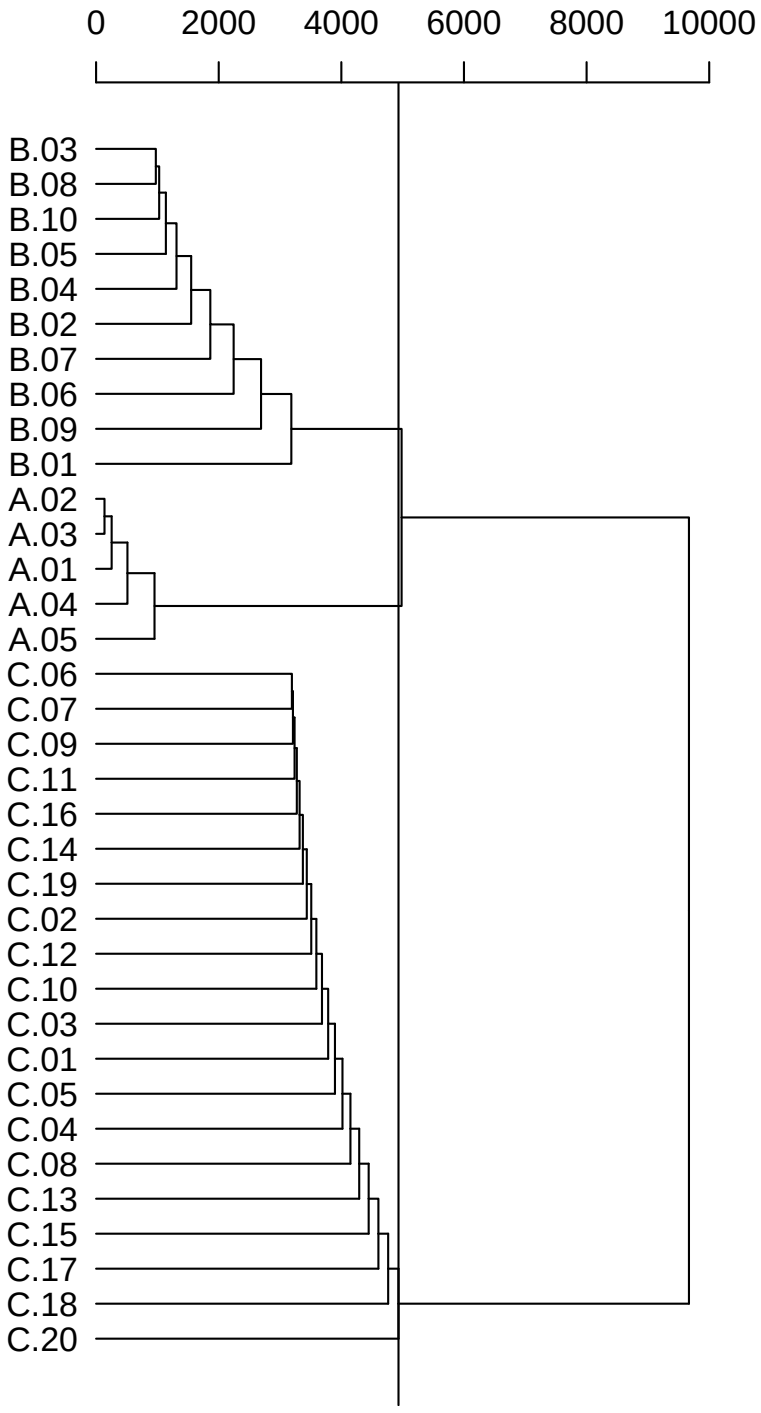

Figure 3.17: Dendrogram of the image data created using the Gaussian variable selection model. The horizontal line shows the optimal grouping proposed by agglomerative method.

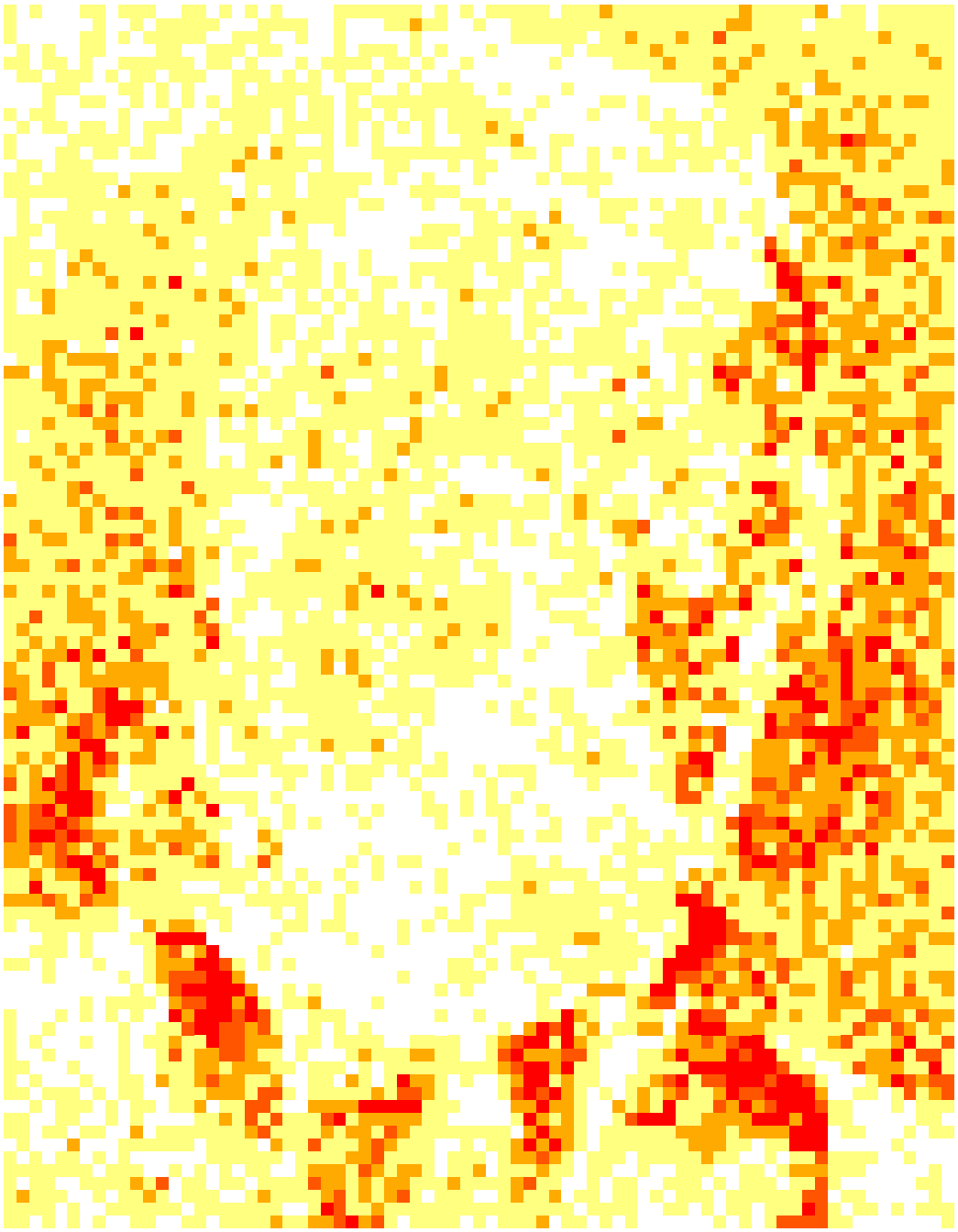

Figure 3.18: Image plot of the $\log B_v^{10}$ using the Gaussian variable selection model. For more details on the heat bar colours, see the caption to Figure 3.9.

Chapter 4

Simulation Results

4.1 Introduction

In this chapter we compare our procedures with another popular clustering procedure, MCLUST. MCLUST is chosen because it is the only clustering method that is automatic, fast and already implemented and documented in R. We perform a Monte Carlo simulation study and compare the procedures using loss functions. Rand (1971) and Binder (1978, 1981) discussed different loss functions appropriate for clustering and Meila (2005) characterises them using mathematical axioms. We use two loss functions, one is a trivial loss (Binder, 1981), which is zero if the estimated clustering equals the true clustering, and one otherwise. The trivial loss function counts how often a clustering method gives wrong partitions, but does not show how bad the wrong partitions are, so we also use another function, proposed by Rand (1971) and Lau and Green (2007), called the misclassification loss.

If the estimated vector of labels, $\hat{\mathbf{d}}$, of length T is composed of elements $\hat{d}_t$, and the vector of true labels $\mathbf{d}$ is composed of elements d_t, this is the same notation as in Section 3.1.1, then the misclassification loss is defined as

$$L(\mathbf{d}, \hat{\mathbf{d}}) = \sum_{t=2}^{T} \sum_{t'<t} I(d_t = d_{t'}, \hat{d}_t \neq \hat{d}_{t'}) + I(\hat{d}_t = \hat{d}_{t'}, d_t \neq d_{t'}),$$

taking values between zero and $T(T-1)/2$. The misclassification loss is zero if the grouping imposed by vector label $\hat{\mathbf{d}}$ is the same as $\mathbf{d}$ and takes the

maximum value $T(T-1)/2$ if the worst mistake happens: the true labels are all the same but mistakenly estimated all different, or vice versa. The trivial loss may be derived from the misclassification loss by composition of the sign function with the misclassification loss, that is, if $L(\mathbf{d}, \hat{\mathbf{d}})$ is the misclassification loss, the trivial loss is $\text{sign}\{L(\mathbf{d}, \hat{\mathbf{d}})\}$.

The effectiveness of clustering methods is studied under different settings motivated by our metabolite example of Section 1.3.1, based on 1000 Monte Carlo replications. The simulated number of non-empty clusters is uniformly distributed between 2 and 10. We excluded the case that all data lie in a single cluster, because it is uninteresting in practice.

This chapter is organised as follows. In Section 4.2 we discuss the simulation results from the Gaussian effects data. Section 4.3 describes the performance of clustering procedures with asymmetric Laplace data, simulated as in the Gaussian effects case, but with the true effects θ_{vc} following an asymmetric Laplace distribution with left-tail variance $\sigma^2_{\theta_L}$ and right-tail variance $\sigma^2_{\theta_R}$, yielding an asymmetric Laplace distribution with variance $\sigma^2_\theta = \sigma^2_{\theta_L} + \sigma^2_{\theta_R}$. In Section 4.4 the quality of parameter estimation is studied for Gaussian and asymmetric Laplace effects data.

It is important to investigate the robustness of our clustering procedures when the assumed model is wrong. Hence, in Section 4.5, the distribution of experimental errors η_{vct} and measurement error ε_{vctr} is chosen to be Student's t with 5 degrees of freedom and scaled to have variances σ^2_η and σ^2, respectively.

Section 4.6 studies the performance of clustering procedures when fundamental assumptions, like independence of variables, existence of experimental error layer η_{vct}, and having replicated observations, are violated. We also briefly discuss the effectiveness of $\log B^{10}_v$ as a measure of importance of variables.

Finally, Section 4.7 summarises important results.

We produce various versions of our proposed clustering procedures by taking different strategies in the parameter estimation step; see Table 4.1 for details.

As competitor for our approaches, we choose the MCLUST procedure of Fraley and Raftery (2002). This is one of the most widely-used Bayesian

Ordinary Procedures		
Fixed Parameters	Gaussian	Laplace
None	G_{vs}	L_{vs}
$q = 1$	G	L
$q = 1, p = 0.01$	$G_p^{0.01}$	$L_p^{0.01}$
$q = 1, p = 0.05$	$G_p^{0.05}$	$L_p^{0.05}$
$q = 1, p = 0.10$	$G_p^{0.10}$	$L_p^{0.10}$
Oracle Procedures		
Description	Gaussian	Laplace
All parameters fixed to the true values.	G_{vs}^*	L_{vs}^*
Parameter $q = 1$, and p is tuned to the simulated value of pq, other parameters are fixed to the true values.	G^*	L^*

Table 4.1: Fitting procedure notation.

clustering procedures, which is also fully automatic and is used for analysis of similar data (Yeung *et al.*, 2001; Dasgupta and Raftery, 1998; Sand and Moore, 2001). MCLUST fails when $p > n$. Hence, as its authors proposed in the software manual, we apply the method to the data projected using principal components, and we denote this in the tables by M. We choose two principal components, but conclusions for another number of principal components are not very different.

Loss values in our simulations are compared pairwise for each simulated setting using non-parametric procedures. Since all the methods are applied on a single dataset in each Monte Carlo replication, a paired test is required for a valid analysis. We propose McNemar's test, a paired test for binary data, for comparison of clustering procedures using the trivial loss, and the Wilcoxon signed-rank test for comparisons using the misclassification loss.

The results of the significance tests are coded in a square matrix with clustering procedures in rows and columns; see Figure 4.2 for an example.

Colours are utilised to code pairwise significances, and symbols are applied to represent the preference respect to the trivial loss; see Tables 4.2 and 4.3 for details. According to the symbols defined in Table 4.3, the best

Description	Colour
McNemar's and the Wilcoxon signed-rank tests are significant at 0.05 level	Orange
McNemar's or the Wilcoxon signed-rank test is significant at 0.05 level	Yellow
McNemar's and the Wilcoxon signed-rank tests are insignificant at 0.05 level	White
McNemar's or the Wilcoxon signed-rank test statistic cannot be calculated	Gray
No comparison is made	Black

Table 4.2: Colours applied for coding pairwise tests, see Figure 4.2.

Description	Sign
The estimated loss is smaller	+
The estimated loss is bigger	−
The estimated loss is equal	.

Table 4.3: Symbols representing preference of the procedures according to the trivial loss, compared with the procedure on the main diagonal, see Figure 4.2.

method, in terms of the trivial loss, is the method on the main diagonal with minus signs on its right side of the same row, and plus signs above it in the same column. The worst clustering procedure is the method with plus signs on its right side and minuses above. The lower triangular part of the matrix is filled with p-values for the Wilcoxon signed-rank test, if they exceed 0.01. The Wilcoxon signed-rank test, unlike McNemar's test, can almost always be calculated. Having p-values in the lower-triangular matrix shows the significance of two methods according to the misclassification loss, especially when McNemar's test statistic cannot be calculated, that is, when the corresponding box is gray.

4.2 Gaussian Effects Model

In this section we study the effectiveness of different procedures applied to data simulated from the Gaussian effects model. The model parameters, $\mu = 0$ and $\sigma^2 = 1$, are relatively easy to estimate. Hence, clustering methods are compared for different values of σ_θ^2, σ_η^2, p, and q. An increase in $\sigma_\theta^2/\sigma_\eta^2$ and in pq gives more clustering information. The hyper-parameter q is the proportion of active variables, and p is the proportion of active variable-cluster combinations for active variables, which gives an expected total of pq active variable-cluster combinations. Datasets are simulated when lots of variables, $q = 0.9$, and small number of variable-cluster combinations for active variables, $p = 0.1$, are activated, giving an expected 9% active variable-clusters. A small proportion of active variables, $q = 0.1$, but large active variable-cluster combinations for active variables, $p = 0.9$ also is considered. In another setting, we fix $p = q = 0.5$, giving 25% variable-cluster combinations active.

In order to compare methods, values for the other hyper-parameters are chosen as follows. The true effect variance, σ_θ^2, is set to 1 or 10, and the experimental error variance, σ_η^2, to 0.5 or 2, yielding signal-to-noise ratio $\sigma_\theta^2/\sigma_\eta^2$ ranging from 0.5 to 40. The trivial and the misclassification losses are represented in Table 4.4. Because we have a total of 40 observations to each cluster, 10 types each with 4 replicates, the maximum value for misclassification loss is $40 \times 39/2 = 780$, whereas for the trivial loss this is 1.

According to Table 4.4, when $\sigma_\theta^2/\sigma_\eta^2$ is small, all methods are almost equally efficient. As the signal to noise ratio $\sigma_\theta^2/\sigma_\eta^2$ increases, both losses decrease, often for all clustering procedures. For instance, compare $\sigma_\theta^2 = 1, \sigma_\eta^2 = 0.5$ with $\sigma_\theta^2 = 20, \sigma_\eta^2 = 0.5$. This is reasonable because an increase to $\sigma_\theta^2/\sigma_\eta^2$ gives more clustering information. For large signal-to-noise ratios, like $\sigma_\theta^2 = 20, \sigma_\eta^2 = 0.5$, the difference between our proposed procedures and MCLUST becomes clearer, in terms of both loss functions and in favour of our methods. This suggests that MCLUST applied on principal components is not a good clustering strategy in our settings. Fitting asymmetric Laplace methods (L, L^*, L_{vs}, and L_{vs}^*) is not very different from Gaussian fits (G, G^*, G_{vs}, and G_{vs}^*). The oracle procedures ($G^*, G_{\text{vs}}^*, L^*$, and L_{vs}^*) with

their parameters fixed to the true values are generally better than the corresponding versions with parameters estimated (G, G_{vs}, L, and L_{vs}). For high signal-to-noise ratios like $\sigma_\theta^2 = 20$ and $\sigma_\eta^2 = 0.5$ with parameters estimated more efficiently, the performance of oracle procedures is close to procedures that estimate the parameters. In Table 4.4, for a few cases fitting the asymmetric Laplace model is better than a Gaussian fit even on Gaussian data. For instance, consider $\sigma_\eta^2 = 0.5, \sigma_\theta^2 = 20$, for which L^* is the best method with respect to both losses. Often, fixing hyper-parameter $q = 1$ and p to a value gives a less efficient method, in terms of both losses for Gaussian and asymmetric Laplace fits, compare $G_p^{0.01}, G_p^{0.05}, G_p^{0.10}, L_p^{0.01}, L_p^{0.05}, L_p^{0.10}$ with the other procedures. This suggests that tuning p when $q = 1$ is crucial.

In Table 4.4, standard errors for the misclassification loss are larger than for the trivial loss. The reason for having tiny standard errors for the trivial loss is that this is a Bernoulli variable, giving standard errors proportional to $\sqrt{\pi(1-\pi)}$, where $0 < \pi < 1$ is the probability of finding the true clustering.

Figure 4.2 shows significance tests related to Table 4.4, where $p = 0.1$ and $q = 0.9$. The top three panels correspond to significance tests of $\sigma_\eta^2 = 0.5$, and the bottom panels to significance tests of $\sigma_\eta^2 = 2$. The three vertical panels refer to $\sigma_\theta^2 = 1, \sigma_\theta^2 = 10$, and $\sigma_\theta^2 = 20$ from left to right, respectively. Hence, for example, tests corresponding to $\sigma_\eta^2 = 0.5$ and $\sigma_\theta^2 = 1$ are found in the top left panel, $\sigma_\eta^2 = 0.5$ and $\sigma_\theta^2 = 10$ in the top middle and top right panel and so forth.

For small signal-to-noise ratio, for example $\sigma_\eta^2 = 2$ and $\sigma_\theta^2 = 1$, the bottom left panel of Figure 4.2, contains gray boxes, since McNemar's test statistic cannot be calculated. When there is restricted clustering information, all clustering methods fail to detect the true labelling and yield two-by-two crosstabs with off-diagonal elements containing low frequencies, where the distribution of McNemar's test statistic cannot be found. An increase of the signal-to-noise ratio $\sigma_\theta^2/\sigma_\eta^2$ leads to significant pairwise comparisons according to the McNemar and Wilcoxon tests and gives yellow and orange yellow boxes. See for example the bottom right, top middle, and the top right panels.

The asymmetric Laplace methods are sometimes better than Gaussian fits but are not significantly preferable to all Gaussian procedures. For example

see $\sigma_\eta^2 = 2, \sigma_\theta^2 = 20$ in Figure 4.2, the top right panel, where L^* is the best method and insignificantly different from G^* in terms of misclassification loss.

Above we discussed situations with a large expected proportion of active variables, $q = 0.9$, and a small expected proportion of active variable-cluster combinations, $p = 0.1$. It is possible to have the same amount of active variable-cluster combinations for data but with a small proportion of active variables, for example by taking $q = 0.1$ and $p = 0.9$; hence, Tables 4.4 and 4.5 are comparable, because they have equal amounts of clustering information on average. The loss values for $q = 0.1$ and $p = 0.9$, especially for extremely large signal-to-noise ratio ($\sigma_\eta^2 = 0.5$ and $\sigma_\theta^2 = 20$), are larger than when $q = 0.9$ and $p = 0.1$, confirming the effectiveness of our method when a lot of variables but few variable-cluster combinations are active. Fitting procedures that mistakenly assume $q = 1$, (G, G^*, L, and L^*), give loss values close to when q is estimated from data (G_{vs} and L_{vs}). This happens because fixing q helps better estimation of the other model parameters and consequently gives better clustering overall. Fixing $q = 1$ may be help toward a better clustering but when $0 < q < 1$ the model measures variable importance, which is demanded in some applications, through B_v^{10}.

Augmenting the amount of clustering information, that is increasing pq from 0.09 to 0.25 ($p = q = 0.5$), gives smaller losses in Table 4.6 than with Tables 4.4 and 4.5 for all cases when $pq = 0.09$. In Table 4.6, we observe that L_{vs} is often less efficient than the corresponding Gaussian procedure, G_{vs}, which shows the impact of low-quality estimation of parameters for the asymmetric Laplace method. Comparing L_{vs}^* with G_{vs}^*, so that the parameter estimation step is removed, both methods perform similarly. The procedures with fixed q and p are often less efficient than methods that estimate p; compare $G_p^{0.01}, G_p^{0.05}$, and $G_p^{0.10}$ with G and G_{vs} for Gaussian models, or $L_p^{0.01}, L_p^{0.05}$, and $L_p^{0.10}$ with L and L_{vs} for the asymmetric Laplace model.

Finally, we note that all of our proposed approaches beat MCLUST implemented on principal components, in terms of both loss functions, for data generated according to the Gaussian effects model. This is not surprising, because Chang (1983) showed that principal components of data are not necessarily informative for clustering. The asymmetric Laplace fits are often as efficient as the Gaussian fits, especially when model parameters are set

to their simulated values (oracle procedures). This suggests, after fixing the parameters to reasonable values, the distribution assumed for the true effects is not so important. The difference between the performance of clustering methods, is caused by the quality of parameter estimation using different distributional assumptions, which we will discuss briefly in Section 4.4. Fixing $q = 1$ and regarding p as a tuning parameter is a good strategy when pq is close to the true value, otherwise it yields a less efficient clustering procedure.

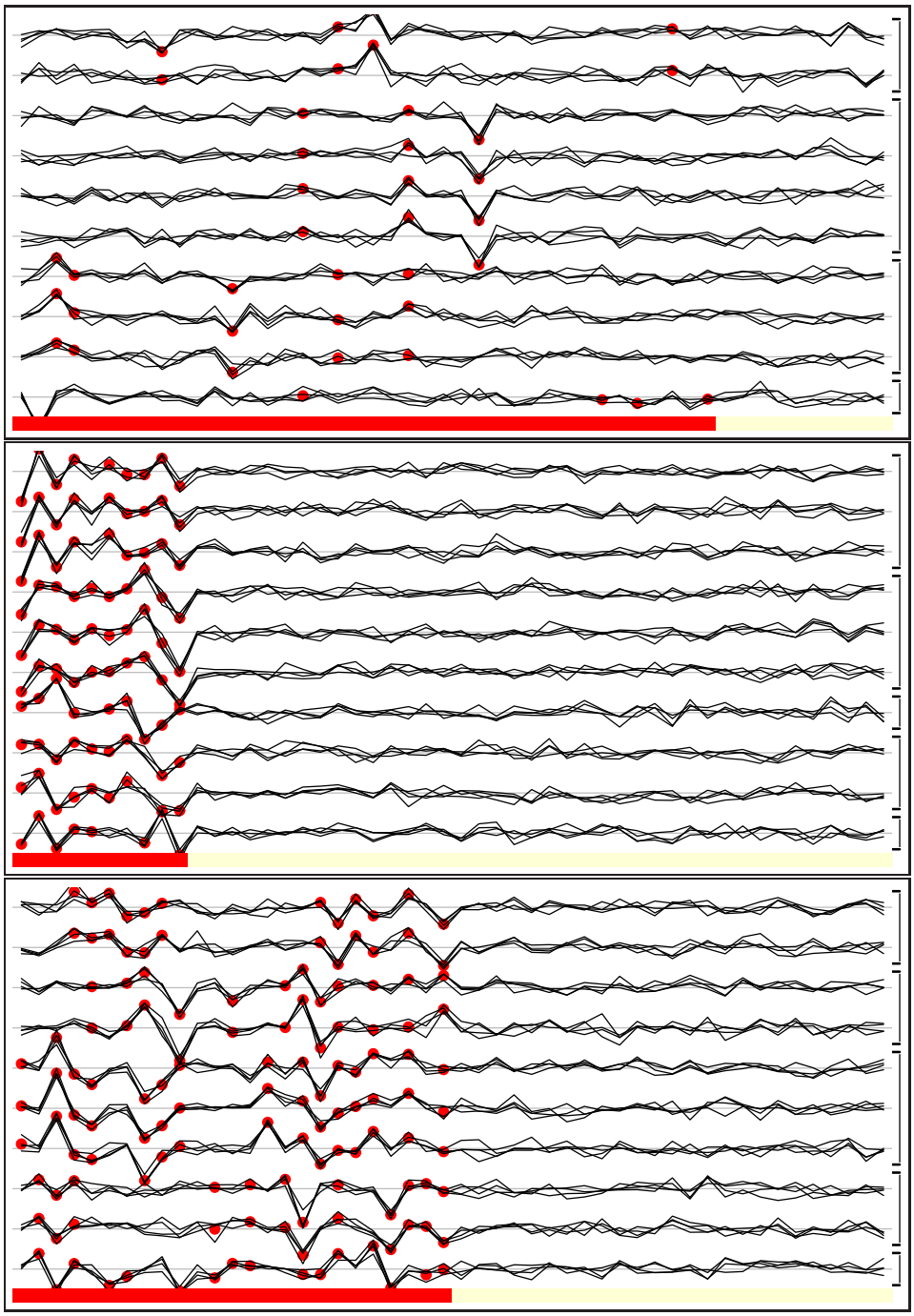

Figure 4.1: Profile plot of simulated Gaussian data with $\sigma_\eta^2 = 0.5$ and $\sigma_\theta^2 = 10$. The hyper-parameters are $p = 0.1, q = 0.9$ for the top panel, $p = 0.9, q = 0.1$, for the middle panel, and $p = q = 0.5$ for the bottom panel. Active variables are represented by a horizontal heat bar at the bottom of each profile plot, red if the variable is active. The active variable-cluster combinations are shown by red solid blobs with probability of appearance equal to p for activated variables. At the right side of each profile plot the simulated grouping is represented.

Loss	Parameter		Fitting Procedure															
	σ_η^2	σ_θ^2	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial(×100)	0.5	1	(0.1) 100	(0.4) 98	(0.4) 98	(0.4) 98	(0.4) 98	(0.3) 99	(0.4) 98	(0.4) 98	(0.4) 98	(0.5) 97	(0.4) 98	(0.5) 97	(0.3) 99	(0.3) 99	(0.3) 99 ”
		10	(1.1) 86	(1.5) 38	(1.5) 40	(1.6) 47	(1.6) 47	(1.6) 54	(1.6) 50	(1.6) 51	(1.5) 38	(1.5) 40	(1.5) 38	(1.6) 57	(1.6) 57	(1.6) 52	(1.6) 53
		20	(1.3) 80	(1.2) 19	(1.3) 20	(1.4) 24	(1.4) 24	(1.4) 25	(1.3) 24	(1.4) 24	(1.2) 18	(1.2) 19	(1.2) 19	(1.5) 32	(1.4) 26	(1.4) 24	(1.4) 24
	2	1	(0.1) 100	(0) 100	(0.6) 97	(0.5) 98	(0.5) 98	(0) 100	(0) 100	(0) 100	(0) 100	(0.6) 96	(0) 100	(0.6) 96	(0) 100	(0) 100	(0) 100
		10	(0.4) 98	(0.6) 96	(0.6) 96	(0.6) 96	(0.6) 96	(0.5) 97	(0.5) 97	(0.5) 97	(0.6) 96	(0.8) 94	(0.6) 96	(0.8) 94	(0.5) 97	(0.5) 98	(0.5) 98
		20	(0.6) 97	(1.2) 84	(1.1) 85	(1) 88	(1) 88	(0.9) 92	(0.9) 91	(0.9) 91	(1.2) 84	(1.1) 85	(1.2) 84	(1.1) 86	(0.9) 92	(0.9) 92	(0.9) 91
Misclassification	0.5	1	(5.8) 476	(6.1) 532	(7.7) 435	(7.6) 447	(7.6) 447	(5.6) 547	(5.7) 540	(5.7) 542	(6) 537	(7.6) 461	(6) 537	(7.6) 461	(5.5) 552	(5.5) 550	(5.5) 553
		10	(4.4) 161	(1.5) 22	(1.7) 26	(2) 34	(2) 34	(2.5) 48	(2.1) 39	(2.2) 42	(1.5) 22	(2.1) 29	(1.5) 22	(5.6) 98	(2.9) 55	(2.6) 46	(2.7) 49
		20	(3.6) 116	(0.8) 8	(1) 9	(1.3) 15	(1.3) 15	(1) 12	(1) 12	(1.1) 12	(0.8) 8	(1.2) 11	(0.9) 8	(4.4) 54	(1) 13	(1.1) 12	(1.1) 13
	2	1	(4.1) 250	(4.6) 578	(7.9) 435	(7.3) 483	(7.3) 483	(4.6) 578	(4.6) 576	(4.6) 577	(4.6) 578	(8.3) 379	(4.6) 578	(8.1) 416	(4.6) 578	(4.6) 577	(4.6) 578
		10	(3.5) 171	(5.7) 219	(5.8) 206	(5.8) 211	(5.8) 211	(6.4) 353	(6.3) 322	(6.5) 339	(6) 230	(6.4) 209	(6) 229	(8) 302	(6.5) 369	(6.6) 346	(6.6) 356
		20	(3.2) 141	(3.1) 94	(3.4) 99	(3.4) 108	(3.4) 108	(4.1) 153	(3.7) 135	(3.9) 142	(3.1) 93	(3.7) 101	(3.1) 94	(6.6) 183	(4.6) 170	(4.2) 148	(4.3) 153

Table 4.4: Losses for simulated data from the Gaussian effects model with parameters $\mu = 0, \sigma^2 = 1, p = 0.1$, and $q = 0.9$. The average of the misclassification and the trivial (×100) losses over 1000 Monte Carlo replications are reported, with the standard error in parentheses above each average loss.

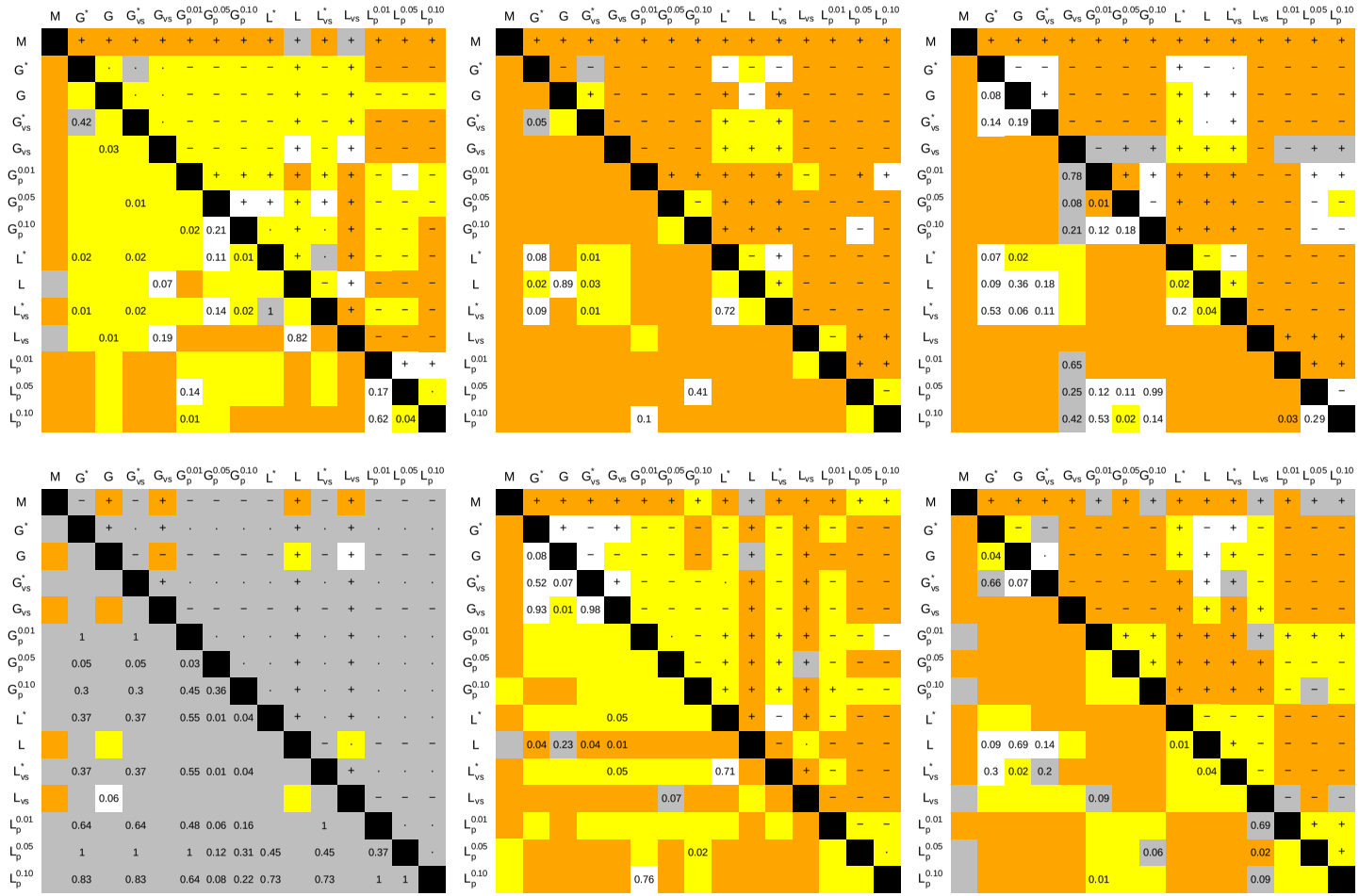

Figure 4.2: Significance test results using McNemar's test for the trivial loss, and the Wilcoxon signed-rank test for the misclassification loss coded in a matrix, for $\mu = 0, \sigma^2 = 1, p = 0.1$, and $q = 0.9$, corresponding to Table 4.4. The top panels refer to $\sigma_\eta^2 = 0.5$ and the bottom panels to $\sigma_\eta^2 = 2$. Significance tests of $\sigma_\theta^2 = 1, 10$ and 20 are positioned on the left, middle and right sides, respectively. Clustering methods are compared pairwise; when no significance at the 0.05 level for both losses is achieved, the corresponding box is white; if one of the tests is significant, it is yellow; if both are significant, orange; and when the p-value could not be calculated for the McNemar's test or the Wilcoxon test, it is gray. Plus, minus and dot correspond to the preferences of each method, according to the trivial loss, compared with the method on the main diagonal; plus for smaller loss (preferable), minus for bigger (undesirable), and dot for equal. The p-values greater than 0.01 for the Wilcoxon signed-rank test are given in the lower-triangular part of the matrix. The figure corresponds to Table 4.4.

Loss	Parameter		Fitting Procedure														
	σ_η^2	σ_θ^2	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times$100)($\times$100)	0.5	1	(0.2) 100	(0.4) 99	(0.5) 97	(0.5) 98	(0.5) 98	(0.3) 99	(0.4) 99	(0.4) 99	(0.3) 99	(0.5) 98	(0.4) 99	(0.3) 99	(0.3) 99	(0.3) 99	(0.3) 99
		10	(1) 88	(1.5) 61	(1.5) 61	(1.6) 59	(1.6) 59	(1.4) 73	(1.5) 70	(1.5) 69	(1.5) 61	(1.6) 60	(1.6) 56	(1.5) 62	(1.4) 74	(1.4) 71	(1.5) 70
		20	(1.3) 77	(1.5) 39	(1.5) 40	(1.6) 44	(1.6) 44	(1.6) 48	(1.6) 45	(1.6) 45	(1.5) 39	(1.5) 39	(1.6) 41	(1.6) 45	(1.6) 49	(1.6) 46	(1.6) 45
	2	1	(0) 100	(0) 100	(0.6) 97	(0.6) 96	(0.6) 96	(0.1) 100	(0.1) 100	(0) 100	(0) 100	(0.7) 95	(0) 100	(0.5) 98	(0) 100	(0) 100	(0) 100
		10	(0.4) 99	(0.5) 98	(0.6) 96	(0.7) 95	(0.7) 95	(0.4) 98	(0.5) 98	(0.5) 97	(0.5) 97	(0.6) 96	(0.6) 96	(0.7) 94	(0.4) 98	(0.5) 98	(0.5) 98
		20	(0.7) 95	(0.9) 92	(0.9) 91	(1) 90	(1) 90	(0.7) 95	(0.7) 95	(0.7) 95	(0.9) 92	(0.9) 91	(1) 88	(1) 89	(0.7) 95	(0.7) 95	(0.7) 95
Misclassification	0.5	1	(5.9) 473	(6.5) 505	(7.8) 423	(7.6) 441	(7.6) 441	(5.6) 545	(6) 528	(5.9) 532	(6.4) 512	(7.6) 461	(6.6) 515	(6.7) 514	(5.5) 551	(5.7) 543	(5.7) 548
		10	(4.6) 148	(3) 60	(3) 61	(2.9) 55	(2.9) 55	(3.1) 78	(3.1) 72	(3.2) 74	(3) 59	(3.3) 62	(2.9) 51	(4.3) 75	(3.3) 82	(3.4) 77	(3.5) 78
		20	(3.4) 82	(2.4) 33	(2.3) 33	(2.4) 37	(2.4) 37	(2.4) 38	(2.4) 36	(2.4) 37	(2.4) 32	(2.6) 35	(2.3) 35	(3.2) 45	(2.5) 40	(2.6) 38	(2.7) 39
	2	1	(4) 249	(4.4) 583	(7.7) 457	(7.5) 475	(7.5) 475	(4.4) 583	(4.5) 581	(4.4) 582	(4.4) 584	(8.3) 390	(4.4) 584	(7.5) 478	(4.5) 582	(4.5) 582	(4.5) 583
		10	(3.4) 156	(6.4) 237	(6.2) 222	(6.4) 198	(6.4) 198	(6.4) 324	(6.4) 303	(6.7) 322	(6.5) 243	(6.9) 236	(6.5) 201	(8.2) 302	(6.5) 346	(6.7) 325	(6.9) 341
		20	(3.2) 119	(4.2) 129	(4.3) 129	(4) 120	(4) 120	(4.5) 164	(4.4) 156	(4.7) 164	(4.2) 129	(4.7) 134	(4.1) 118	(5.9) 159	(4.9) 178	(4.8) 165	(5.1) 174

Table 4.5: Losses and their respective standard errors in parentheses above each estimated loss, for data generated under the Gaussian effects model with parameters $\mu = 0, \sigma^2 = 1, p = 0.9$, and $q = 0.1$. See the caption to Table 4.4 for details.

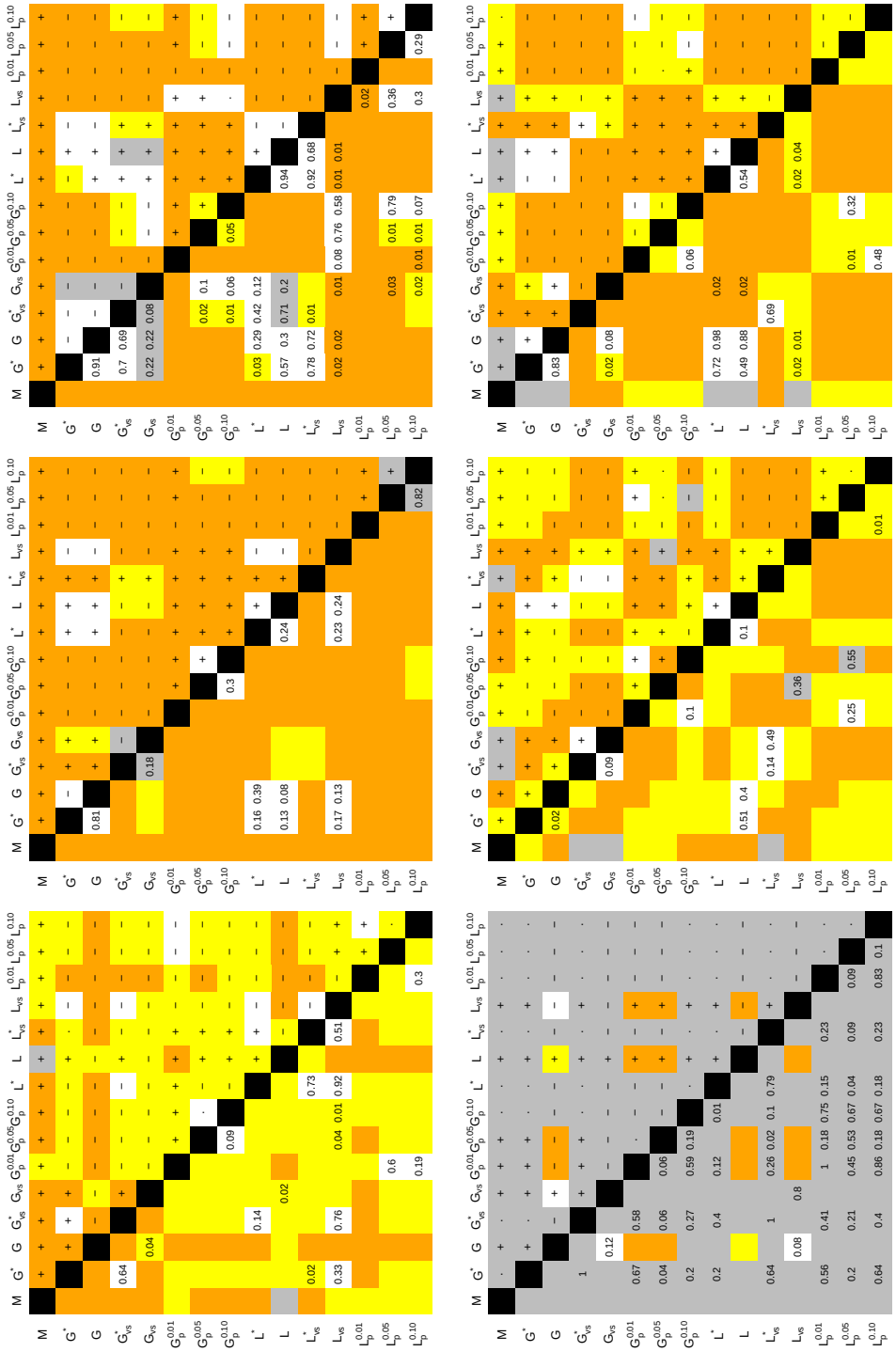

Figure 4.3: Significance test results for data generated by the Gaussian effects model with $\mu = 0, \sigma^2 = 1, p = 0.9$, and $q = 0.1$, corresponding to Table 4.5. For details see the caption to Figure 4.2.

Loss	Parameter		Fitting Procedure														
	σ_η^2	σ_θ^2	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times 100$)	0.5	1	(0.5) 98	(0.8) 93	(0.8) 94	(0.7) 96	(0.7) 96	(0.5) 98	(0.6) 97	(0.6) 96	(0.8) 94	(0.8) 93	(0.7) 94	(0.5) 97	(0.4) 98	(0.5) 97	(0.5) 97
		10	(1.4) 74	(0.7) 4	(0.7) 5	(1) 11	(1) 11	(1.6) 46	(1.5) 33	(1.5) 32	(0.6) 4	(0.9) 10	(0.8) 7	(1.5) 32	(1.6) 52	(1.5) 39	(1.5) 37
		20	(1.5) 63	(0.3) 1	(0.3) 1	(0.7) 5	(0.7) 5	(1.2) 18	(1) 11	(0.9) 10	(0.3) 1	(0.4) 2	(0.5) 3	(1.1) 15	(1.3) 23	(1.1) 16	(1.1) 14
	2	1	(0) 100	(0) 100	(0.7) 95	(0.5) 97	(0.5) 97	(0) 100	(0) 100	(0) 100	(0) 100	(0.8) 94	(0) 100	(0.5) 97	(0) 100	(0) 100	(0) 100
		10	(0.8) 94	(1.3) 78	(1.3) 79	(1.2) 82	(1.2) 82	(0.6) 96	(0.7) 94	(0.8) 94	(1.3) 78	(1.1) 84	(1.3) 78	(0.9) 90	(0.6) 96	(0.7) 95	(0.7) 95
		20	(1) 87	(1.5) 68	(1.5) 69	(1.3) 77	(1.3) 77	(1.1) 84	(1.3) 78	(1.3) 78	(1.5) 68	(1.3) 77	(1.4) 75	(1.2) 84	(1.1) 87	(1.2) 82	(1.3) 80
Misclassification	0.5	1	(6.7) 357	(8.6) 312	(8.1) 277	(8.3) 299	(8.3) 299	(6.6) 520	(7.2) 484	(7.4) 482	(8.8) 357	(8.6) 359	(8.8) 355	(7.5) 487	(6.3) 538	(7) 510	(7) 512
		10	(2.2) 58	(0.6) 3	(0.5) 3	(1.2) 10	(1.2) 10	(1.7) 28	(1.1) 15	(1.3) 16	(0.6) 3	(0.8) 5	(0.9) 5	(4.2) 47	(2.5) 40	(1.7) 23	(1.8) 24
		20	(1.6) 34	(0.2) 0	(0.2) 0	(0.9) 5	(0.9) 5	(0.8) 7	(0.7) 4	(0.7) 4	(0.2) 0	(0.6) 1	(0.6) 2	(1.9) 14	(1.5) 12	(1.2) 7	(1.2) 7
	2	1	(3.8) 223	(4.4) 589	(8.4) 390	(7.6) 478	(7.6) 478	(4.4) 588	(4.5) 586	(4.5) 586	(4.4) 590	(8.5) 341	(4.4) 590	(7.9) 454	(4.5) 587	(4.5) 586	(4.5) 587
		10	(3) 100	(2.9) 74	(3.1) 81	(3.3) 88	(3.3) 88	(6.7) 291	(6.1) 228	(6.3) 234	(3) 74	(3.8) 104	(2.9) 77	(8.1) 262	(7) 324	(6.6) 263	(7) 266
		20	(2.8) 75	(2.6) 63	(2.7) 66	(3.2) 82	(3.2) 82	(4) 115	(3.2) 84	(3.2) 86	(2.6) 63	(3.3) 84	(2.8) 73	(5.8) 143	(4.4) 134	(3.4) 97	(3.4) 95

Table 4.6: Losses and their standard errors in parentheses above each loss, for data generated under the Gaussian effects model with hyper-parameters $\mu = 0, \sigma = 1, p = 0.5$, and $q = 0.5$. See the caption to Table 4.4 for details.

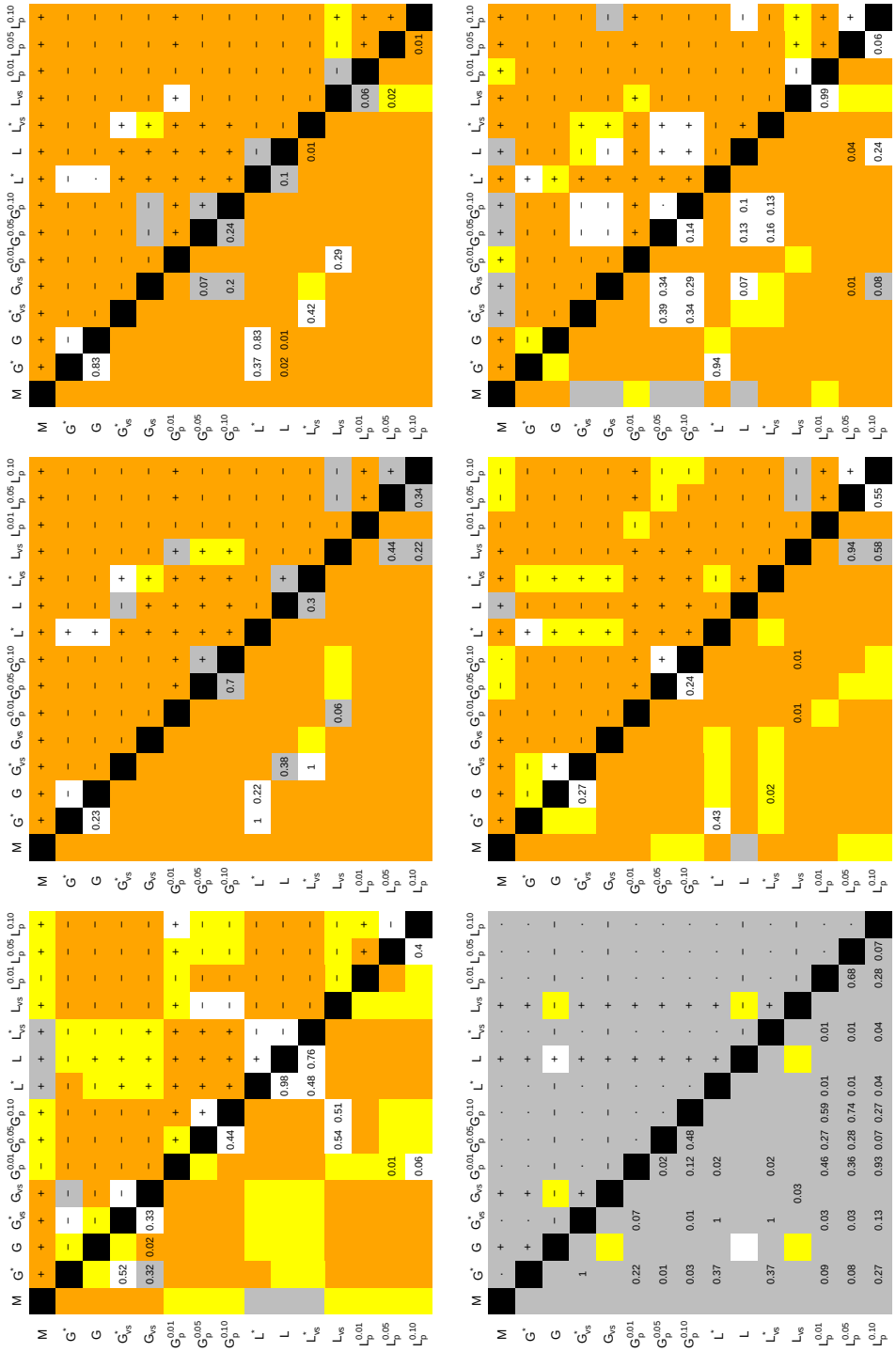

Figure 4.4: Significance test results for data generated under the Gaussian effects model with $\mu = 0, \sigma^2 = 1, p = 0.5$, and $q = 0.5$, corresponding to Table 4.6. For details see the caption to Figure 4.2.

4.3 Asymmetric Laplace Effects Model

The asymmetric Laplace data are generated like the Gaussian data described in Section 4.1, except that the true effects θ_{vc} are generated from asymmetric Laplace distribution with variance $\sigma_\theta^2 = \sigma_{\theta_L}^2 + \sigma_{\theta_R}^2$, where $\sigma_{\theta_L}^{-1}$ is the left-tail rate and $\sigma_{\theta_R}^{-1}$ is the right-tail rate of the exponential distributions comprising the Laplace law. The ratio $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2$ measures the skewness. When the ratio equals one it yields the symmetric Laplace (the double exponential) distribution and when $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$ it leads to a right-skewed distribution, see Figure 2.5. Profile plots of data simulated from the asymmetric Laplace model with $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10, \sigma_\theta^2 = 10, \sigma_\eta^2 = 0.5$ are given in Figure 4.5. Most peaks in the profile plots are positive, because the effects are simulated from a skewed distribution ($\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$) with a higher probability of having positive effects.

Table 4.7 is comparable with Table 4.4, Table 4.8 with Table 4.4, and Table 4.9 with Table 4.6. Comparing the mentioned tables we observe that it is harder to find the true clustering when effects are distributed according to a symmetric Laplace distribution ($\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 1$) with the same variance. It is even more difficult when the effects are asymmetric ($\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$), because the true effects are more concentrated about zero. However, asymmetric Laplace fits are similar to Gaussian fits on asymmetric Laplace data, even if effects are highly asymmetric ($\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$). The asymmetric Laplace fit L_{vs} suffers from poor parameter estimation, yielding greater loss values than its corresponding Gaussian fit G_{vs}. In such cases, knowing the true parameters gives better performance; compare L^* with L and L_{vs}^* with L_{vs} in Table 4.7. In Figures 4.6 and 4.7, similar regions are yellow, orange and white, confirming that asymmetric effects do not change the behaviour of our clustering procedures. In Table 4.9 we see that the estimated losses decrease for $pq = 0.25$ compared with cases that have fewer active variable-cluster combinations (Tables 4.7 and 4.8 having $pq = 0.09$). Asymmetric effects have little effect on clustering performance, especially when there is moderately strong clustering information, for example in Table 4.9 compare $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 1$ with $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$ when $\sigma_\theta^2 = 20$.

We conclude that data generated with the asymmetric Laplace effects

model mostly follow a similar pattern as for the Gaussian effects models discussed in Section 4.2. This confirms that a wrong assumption about the mixing distribution does not change the result of clustering. In another words, having a mixture model is more important than the exact distribution of the effects, confirming the results of Bhowmick *et al.* (2006).

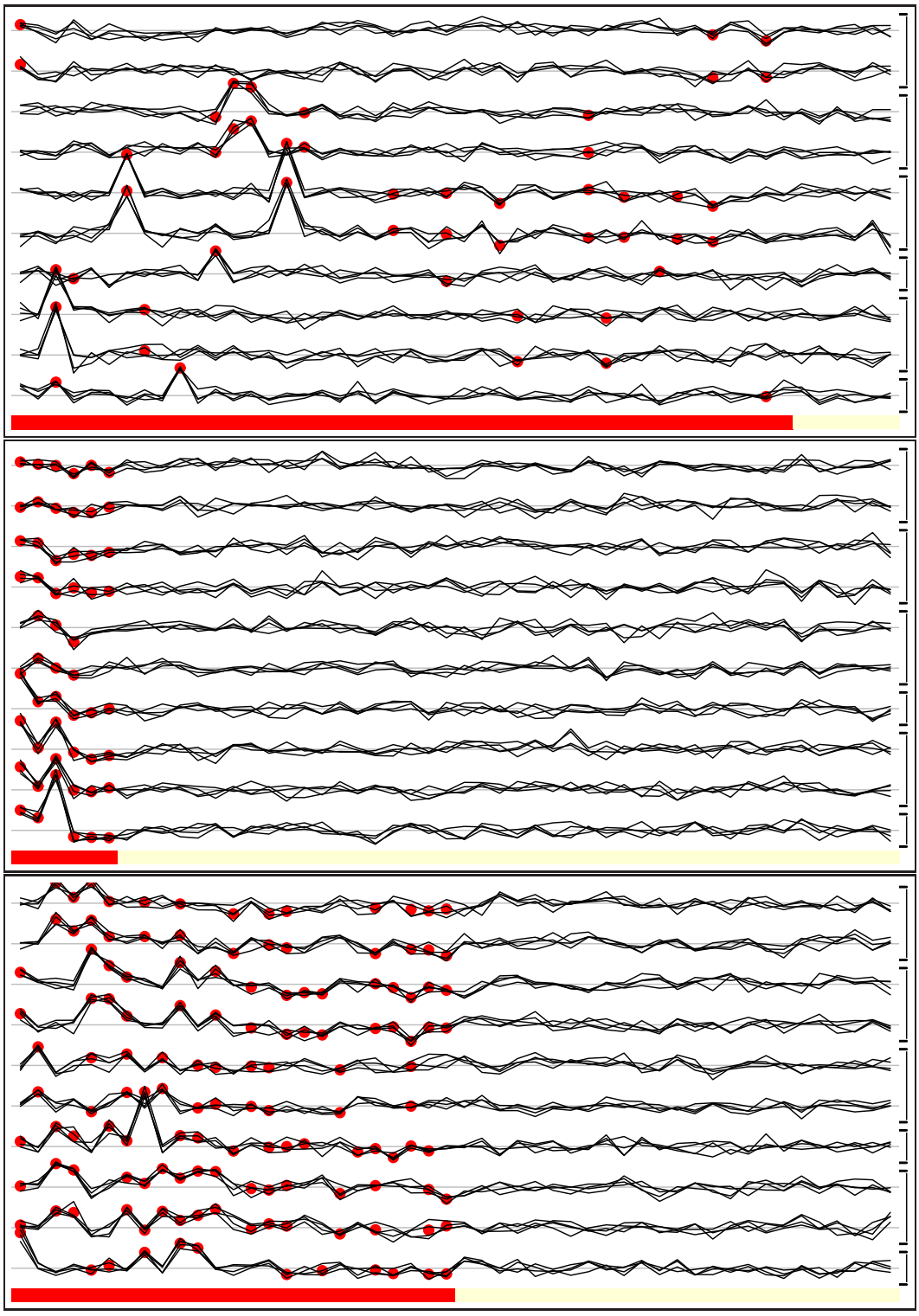

Figure 4.5: Profile plot of data generated from the asymmetric Laplace effects model with $\sigma_\eta^2 = 0.5$, $\sigma_\theta^2 = \sigma_{\theta_R}^2 + \sigma_{\theta_L}^2 = 10$ and $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$. The hyper-parameters are $p = 0.1$ and $q = 0.9$ (the top panel) $p = 0.9$ and $q = 0.1$ (the middle panel) and $p = q = 0.5$ (bottom panel). For more details see Figure 4.1.

Loss	Parameter		Fitting Procedure														
	$\sigma^2_{\theta_R}/\sigma^2_{\theta_L}$	σ^2_θ	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times 100$)	1	1	(0.2) 100	(0.5) 98	(0.5) 98	(0.5) 98	(0.5) 98	(0.4) 98	(0.4) 98	(0.4) 98	(0.5) 98	(0.5) 98	(0.5) 98	(0.4) 98	(0.3) 99	(0.4) 98	(0.4) 99
		10	(1) 89	(1.6) 57	(1.6) 58	(1.6) 57	(1.5) 64	(1.5) 66	(1.5) 64	(1.5) 64	(1.6) 57	(1.6) 58	(1.6) 57	(1.4) 76	(1.5) 67	(1.5) 64	(1.5) 65
		20	(1.2) 82	(1.5) 33	(1.5) 35	(1.5) 33	(1.5) 39	(1.6) 41	(1.6) 40	(1.6) 41	(1.5) 32	(1.5) 34	(1.5) 32	(1.6) 57	(1.6) 42	(1.6) 40	(1.6) 41
	10	1	(0.2) 100	(0.5) 98	(0.5) 98	(0.5) 98	(0.4) 98	(0.5) 98	(0.5) 98	(0.4) 98	(0.5) 98	(0.5) 97	(0.5) 98	(0.4) 98	(0.4) 98	(0.4) 99	(0.3) 99
		10	(0.9) 90	(1.5) 66	(1.5) 68	(1.5) 66	(1.4) 72	(1.4) 73	(1.4) 71	(1.4) 72	(1.5) 66	(1.5) 70	(1.5) 67	(1.3) 80	(1.4) 73	(1.4) 72	(1.4) 72
		20	(1.1) 86	(1.6) 49	(1.6) 52	(1.6) 49	(1.6) 56	(1.6) 56	(1.6) 54	(1.6) 56	(1.6) 47	(1.6) 50	(1.6) 47	(1.5) 66	(1.6) 56	(1.6) 54	(1.6) 55
Misclassification	1	1	(5.8) 471	(6.7) 497	(7.5) 412	(6.7) 497	(7.3) 430	(6.2) 504	(6.3) 493	(6.1) 504	(6.6) 499	(7.5) 443	(6.6) 499	(7.5) 459	(6) 516	(6.1) 513	(6) 522
		10	(4.5) 181	(2.5) 47	(2.6) 52	(2.5) 47	(2.7) 59	(3.1) 72	(3) 66	(3) 69	(2.5) 47	(2.7) 51	(2.5) 47	(7.8) 208	(3.3) 77	(3.3) 70	(3.3) 74
		20	(3.8) 136	(1.4) 19	(1.5) 21	(1.4) 19	(1.9) 28	(1.7) 27	(1.7) 26	(1.7) 27	(1.4) 19	(1.7) 22	(1.4) 19	(7.2) 138	(2) 29	(1.8) 27	(1.9) 28
	10	1	(5.7) 461	(6.7) 480	(7.2) 406	(6.7) 480	(7) 422	(6.3) 484	(6.3) 474	(6.2) 481	(6.7) 472	(7.5) 420	(6.7) 472	(7.6) 441	(6) 513	(5.9) 514	(5.8) 525
		10	(4.7) 207	(3.4) 79	(3.5) 86	(3.4) 79	(3.4) 90	(3.9) 106	(3.7) 98	(3.9) 106	(3.4) 77	(4.3) 94	(3.4) 77	(7.9) 218	(4.8) 125	(4.8) 118	(4.9) 122
		20	(3.9) 155	(2.1) 38	(2.2) 42	(2.1) 38	(2.3) 46	(2.5) 50	(2.3) 46	(2.4) 51	(1.9) 35	(2.6) 43	(1.9) 35	(7.3) 154	(2.9) 55	(2.8) 50	(2.8) 52

Table 4.7: Losses and their standard errors in parentheses above each value, for data generated using the asymmetric Laplace effects model with $\mu = 0, \sigma = 1, \sigma^2_\eta = 0.5, p = 0.1$, and $q = 0.9$.

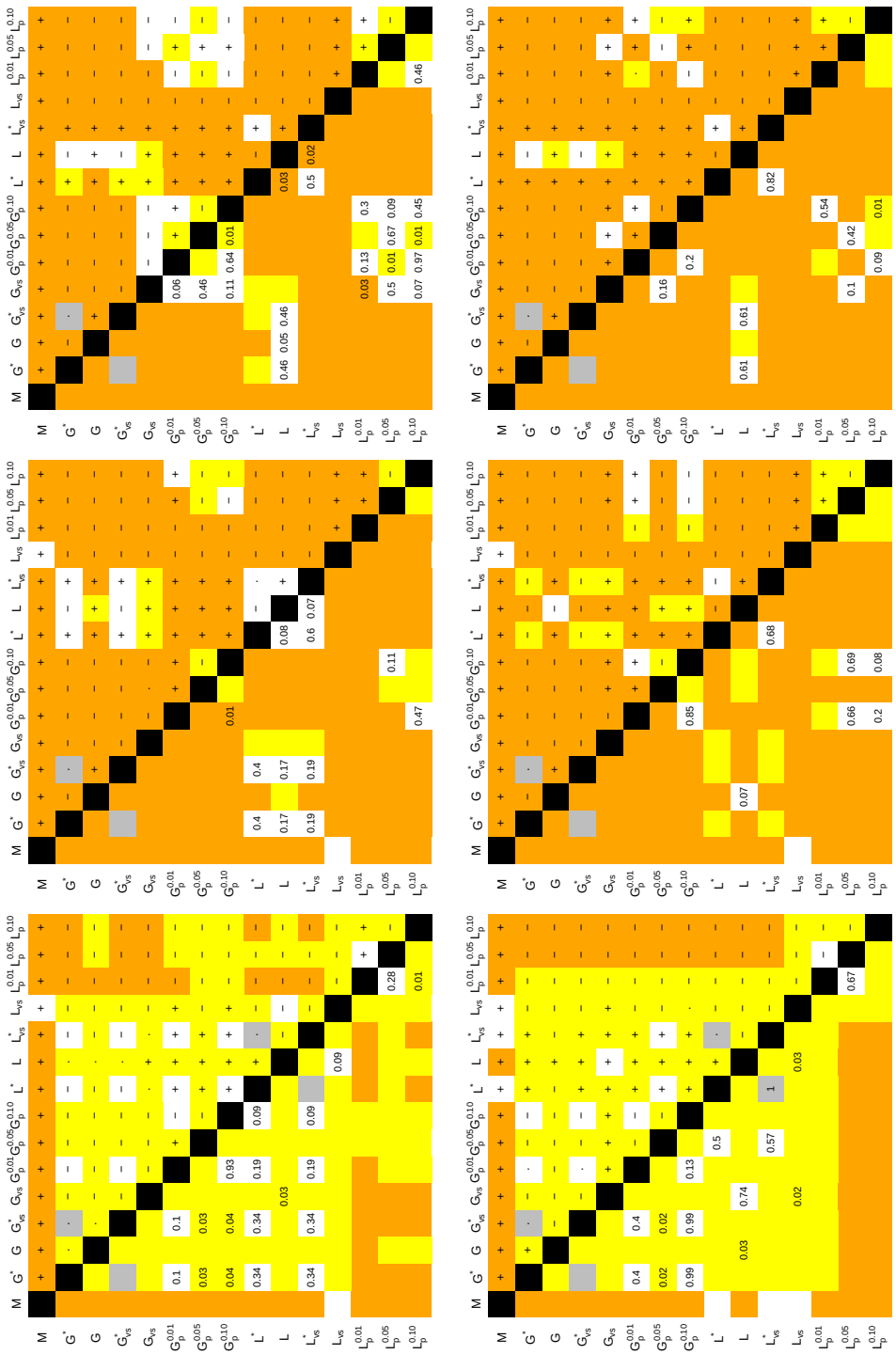

Figure 4.6: Significance tests for data generated using the asymmetric Laplace effects model with parameters $\mu = 0, \sigma^2 = 1, \sigma_\eta^2 = 0.5, p = 0.1$, and $q = 0.9$, corresponding to Table 4.7. The top panels refer to $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 1$ and bottom panels to $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$. The tests corresponding to $\sigma_\theta^2 = 1, 10$ and 20 are on the left, middle and right sides, respectively. The figure corresponds to Table 4.7. See the caption to Figure 4.2 for details.

Loss	Parameter		Fitting Procedure														
	$\sigma^2_{\theta_R}/\sigma^2_{\theta_L}$	σ^2_θ	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times 100$)	1	1	(0.1) 100	(0.4) 98	(0.4) 98	(0.4) 98	(0.4) 99	(0.4) 98	(0.4) 99	(0.4) 98	(0.4) 99	(0.5) 98	(0.3) 99	(0.1) 100	(0.4) 99	(0.4) 98	(0.3) 99
		10	(0.9) 91	(1.4) 76	(1.3) 77	(1.5) 70	(1.4) 76	(1.3) 80	(1.3) 79	(1.3) 80	(1.3) 76	(1.3) 77	(1.4) 74	(1.3) 80	(1.2) 81	(1.2) 81	(1.3) 80
		20	(1.1) 86	(1.6) 58	(1.6) 59	(1.6) 56	(1.6) 59	(1.5) 65	(1.5) 62	(1.5) 62	(1.6) 58	(1.6) 58	(1.6) 56	(1.5) 61	(1.5) 66	(1.5) 63	(1.5) 63
	10	1	(0.2) 99	(0.4) 98	(0.5) 98	(0.4) 98	(0.5) 98	(0.5) 98	(0.5) 98	(0.4) 98	(0.4) 98	(0.5) 98	(0.4) 98	(0.2) 100	(0.4) 98	(0.4) 98	(0.3) 99
		10	(0.8) 92	(1.2) 81	(1.2) 82	(1.3) 77	(1.2) 84	(1.1) 84	(1.2) 83	(1.2) 83	(1.2) 82	(1.2) 83	(1.2) 83	(0.9) 90	(1.1) 85	(1.2) 84	(1.2) 84
		20	(1) 88	(1.5) 69	(1.4) 70	(1.5) 63	(1.5) 70	(1.4) 73	(1.4) 71	(1.4) 71	(1.5) 68	(1.4) 70	(1.5) 68	(1.3) 77	(1.4) 73	(1.4) 72	(1.4) 72
Misclassification	1	1	(5.6) 477	(6.3) 514	(7.2) 440	(6.3) 514	(6.8) 474	(6.1) 507	(6.1) 503	(5.9) 518	(6.3) 512	(7.1) 467	(6.2) 517	(6.2) 523	(5.9) 515	(6) 516	(5.8) 529
		10	(5.5) 222	(5) 117	(4.7) 117	(4.9) 107	(4.1) 97	(5.4) 144	(5.4) 138	(5.7) 147	(4.9) 117	(5) 120	(4.1) 92	(6.3) 151	(5.6) 150	(5.6) 146	(5.9) 152
		20	(4) 139	(3.2) 59	(3) 58	(3.1) 56	(2.9) 56	(3.8) 75	(3.7) 70	(3.8) 73	(3.2) 59	(3.2) 59	(2.8) 52	(4.1) 73	(4.1) 80	(3.9) 73	(4) 75
	10	1	(5.6) 477	(6.4) 499	(7.2) 420	(6.4) 499	(7) 444	(6.3) 478	(6.4) 476	(6.2) 494	(6.5) 489	(7.3) 451	(6.7) 489	(5.9) 535	(6.1) 503	(6.1) 510	(5.8) 527
		10	(5.7) 245	(5.3) 148	(5.1) 147	(5.4) 142	(4.6) 129	(5.5) 168	(5.5) 165	(5.8) 175	(5.3) 145	(5.7) 158	(4.8) 126	(8.5) 344	(5.9) 180	(6) 177	(6.3) 188
		20	(4.4) 163	(4) 87	(3.8) 86	(4) 81	(3.4) 78	(4) 97	(3.9) 92	(4.2) 97	(3.9) 84	(4.3) 93	(3.4) 76	(7.6) 193	(4.4) 104	(4.5) 102	(4.6) 105

Table 4.8: Losses and their standard errors in parentheses above each value, for data generated using the asymmetric Laplace effects model with $\mu = 0, \sigma = 1, \sigma^2_\eta = 0.5, p = 0.9$, and $q = 0.1$.

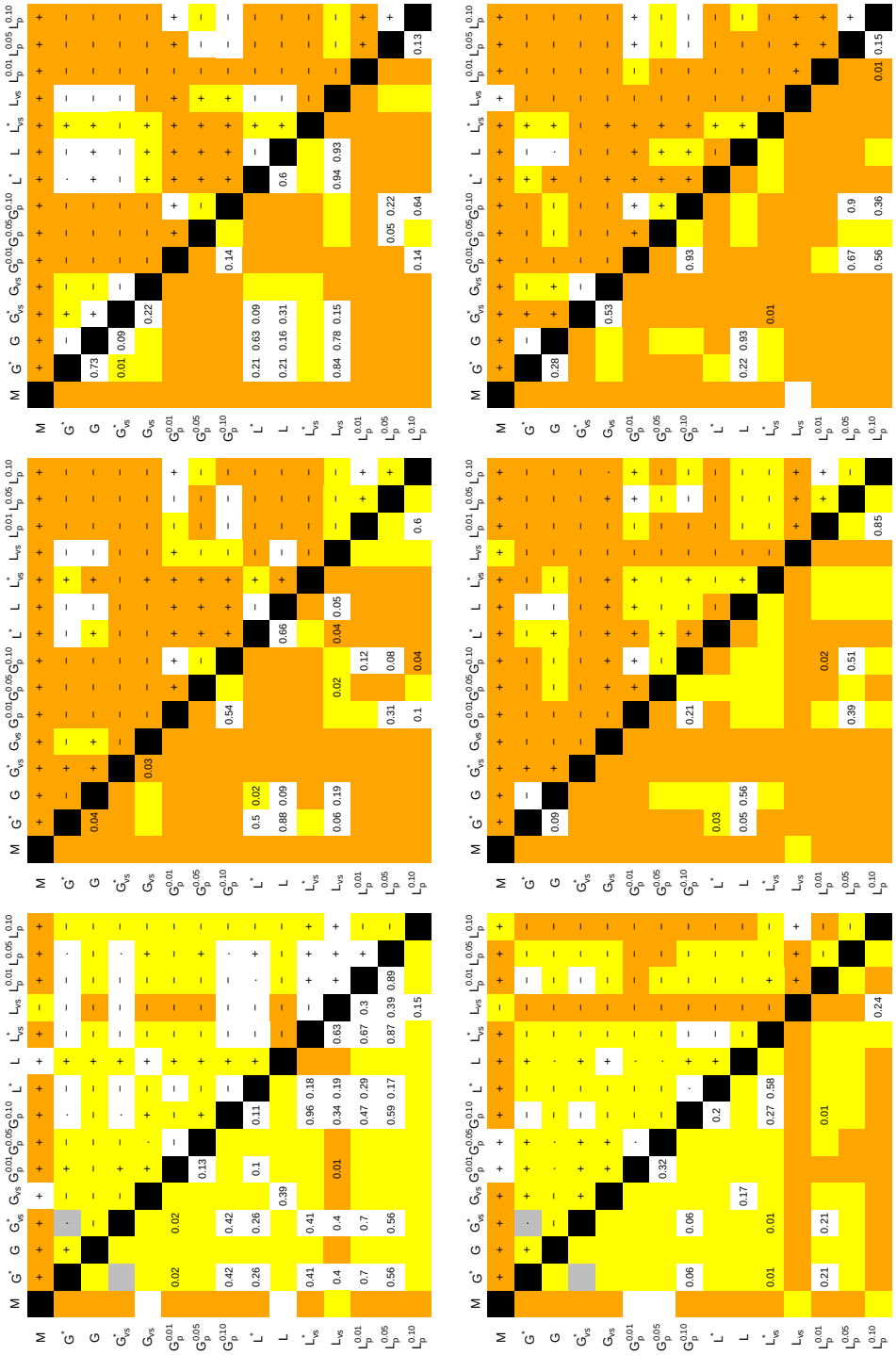

Figure 4.7: Significance tests for data generated using the asymmetric Laplace effects model with $\mu = 0, \sigma^2 = 1, \sigma_\eta^2 = 0.5, p = 0.9$, and $q = 0.1$, corresponding to Table 4.8. For more details see the caption to Figure 4.6.

Loss	Parameter		Fitting Procedure														
	$\sigma^2_{\theta_R}/\sigma^2_{\theta_L}$	σ^2_θ	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times 100$)	1	1	(0.4) 98	(0.7) 95	(0.7) 94	(0.7) 95	(0.6) 96	(0.7) 96	(0.7) 95	(0.7) 95	(0.7) 95	(0.7) 94	(0.7) 95	(0.6) 96	(0.6) 96	(0.7) 96	(0.7) 96
		10	(1.3) 79	(1) 11	(1.1) 14	(1) 13	(1.2) 18	(1.6) 40	(1.5) 34	(1.5) 33	(1) 11	(1) 12	(1) 12	(1.6) 49	(1.6) 42	(1.5) 35	(1.5) 35
		20	(1.4) 73	(0.5) 2	(0.5) 3	(0.8) 7	(0.8) 7	(1.1) 16	(1) 12	(1) 10	(0.5) 2	(0.5) 3	(0.6) 4	(1.5) 37	(1.2) 18	(1) 12	(1) 11
	10	1	(0.4) 98	(0.7) 95	(0.8) 94	(0.7) 94	(0.6) 96	(0.7) 94	(0.8) 94	(0.8) 94	(0.7) 95	(0.7) 95	(0.7) 94	(0.6) 96	(0.6) 96	(0.7) 96	(0.6) 96
		10	(1.3) 80	(1.3) 20	(1.4) 26	(1.3) 21	(1.4) 28	(1.6) 48	(1.6) 40	(1.6) 40	(1.3) 20	(1.3) 22	(1.3) 21	(1.6) 59	(1.6) 50	(1.6) 44	(1.6) 42
		20	(1.4) 72	(0.7) 5	(0.8) 7	(1) 12	(0.9) 9	(1.3) 20	(1.2) 16	(1.1) 16	(0.7) 6	(0.8) 6	(0.8) 8	(1.6) 44	(1.3) 22	(1.2) 16	(1.1) 15
Misclassification	1	1	(6.6) 399	(8.6) 392	(7.8) 301	(8.6) 392	(7.9) 330	(7.3) 435	(7.4) 418	(7.5) 439	(8.6) 400	(8.3) 343	(8.6) 401	(7.4) 494	(7.2) 464	(7.3) 455	(7.3) 472
		10	(3.3) 96	(0.7) 5	(0.7) 5	(0.7) 5	(1.2) 12	(1.3) 18	(1) 13	(1) 13	(0.7) 5	(0.9) 6	(0.9) 7	(8.9) 200	(1.3) 19	(1.1) 14	(1) 14
		20	(2.3) 59	(0.5) 1	(0.5) 2	(0.5) 2	(1) 6	(0.4) 4	(0.5) 4	(0.5) 3	(0.5) 2	(0.5) 2	(0.6) 3	(8.3) 160	(0.4) 5	(0.5) 4	(0.5) 4
	10	1	(6.5) 392	(8.5) 381	(7.2) 284	(8.6) 381	(7.4) 311	(7.2) 394	(7.2) 374	(7.3) 396	(8.5) 357	(8.2) 338	(8.5) 350	(7.7) 463	(7.1) 452	(7.3) 457	(7.1) 480
		10	(3) 96	(0.8) 8	(0.9) 11	(0.8) 8	(1.4) 17	(1.3) 23	(1.2) 18	(1.2) 19	(0.9) 8	(1.3) 12	(1) 10	(8.5) 205	(2.3) 32	(2.6) 30	(2.8) 30
		20	(2.3) 62	(0.5) 2	(0.5) 3	(0.5) 3	(0.9) 6	(0.6) 7	(0.6) 6	(0.6) 6	(0.5) 2	(0.4) 3	(0.7) 4	(8.3) 171	(1) 9	(1) 7	(1) 7

Table 4.9: Losses and their standard errors in parentheses above each value, for data generated using the asymmetric Laplace effects model with $\mu = 0, \sigma = 1, \sigma^2_\eta = 0.5, p = 0.5$, and $q = 0.5$.

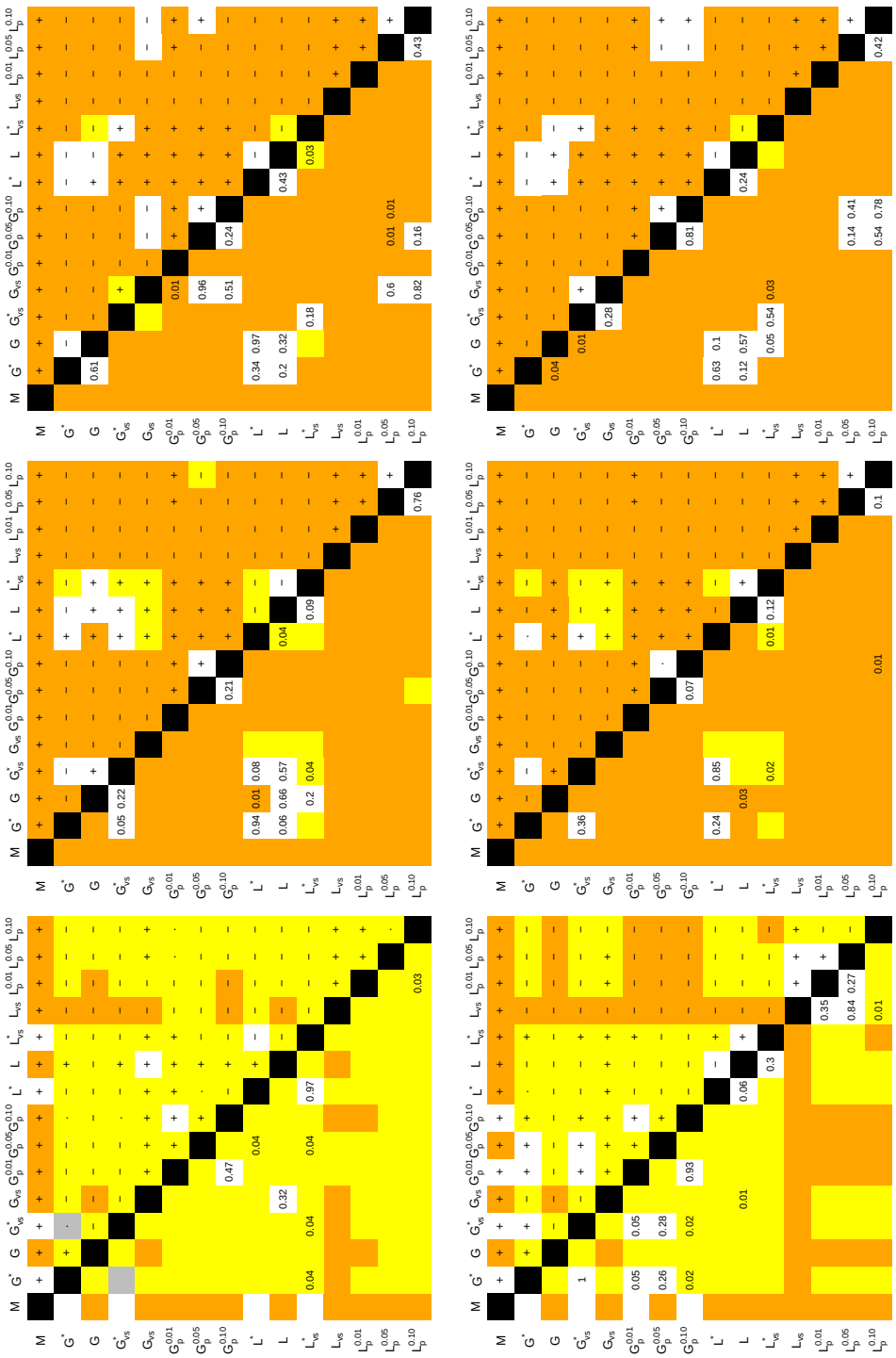

Figure 4.8: Significance tests for data generated using the asymmetric Laplace effects model with $\mu = 0, \sigma^2 = 1, \sigma_\eta^2 = 0.5, p = 0.5$, and $q = 0.5$, corresponding to Table 4.9. For more details see the caption to Figure 4.6.

4.4 Parameter Estimation

The simulation studies of Sections 4.2 and 4.3 show that the efficiency of our proposed approach depends on the model parameters. This section studies the quality of estimation of the parameters using maximum likelihood. We considered our proposed procedures with model parameters fixed to the true values and called them oracle methods, $G^*_{\text{vs}}, L^*_{\text{vs}}, G^*$, and L^*. However, in practice the parameters must be estimated from data. Poor estimation of them often yields poor classification and clustering. In this section we discuss the quality of estimation of parameters using maximum likelihood assuming, initially, that each type is a separate cluster. We considered the Gaussian fit when all parameters are estimated, denoted by G_{vs}; the parameter q is fixed to one and all other parameters are estimated, represented by G; fixing q to one and p to $0.01, 0.05$, and 0.1 are denoted by $G_p^{0.01}, G_p^{0.05}$, and $G_p^{0.10}$, respectively. For the asymmetric Laplace fits, similar versions are considered, denoted by L_{vs}, L, $L_p^{0.01}$, $L_p^{0.05}$, and $L_p^{0.10}$. In all figures, the true values of parameters are shown by a horizontal line and boxplots are used to represent the distribution of the estimated hyper-parameters. White boxplots correspond to data simulated from the Gaussian effects model, yellow and green refer to data generated from symmetric ($\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 1$) and asymmetric Laplace models ($\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 10$), respectively.

This section is organised as follows. First, the estimation of parameters μ and σ^2 is discussed. Then, the quality of estimation of parameter related to proportion of active variables, q, and active variable-cluster combinations for active variables, p, is discussed. The variance for experimental error σ^2_η is studied afterwards. Finally, we study estimation performance for the effects variance, σ^2_θ. For data generated from the asymmetric Laplace model there are two variance parameters, one corresponding to the left tail, $\sigma^2_{\theta_L}$, and another to the right tail, $\sigma^2_{\theta_R}$, of the asymmetric Laplace distribution. We discuss the case $\mu = 0$, $\sigma^2 = 1$ and $\sigma^2_\eta = 0.5$ with other hyper-parameters varied as in Sections 4.2 and 4.3.

In all the simulations mentioned in Section 4.2 and 4.3, the parameter μ is set to zero. The parameter μ is the location parameter of the data when the true effects, θ_{vc}, are removed. It may be simply estimated by taking the mean

or median of data. Hence, for negligible effects, like small values of σ_θ^2, or pq, estimation of μ is easy. Otherwise, the quality of its estimation depends on other estimated parameters, as confirmed in Figure 4.9. For data generated from the Gaussian effects model, when σ_θ^2 and pq are small, estimation of μ is more precise; in Figure 4.9 compare the top left with the top right panel. Generating effects from the asymmetric Laplace distribution does not change the precision of estimation; compare the top panels of Figure 4.9 with its bottom panels. Clearly estimation of μ using the asymmetric Laplace fit is more difficult than with the Gaussian fit; compare the boxplots of G_{vs} with L_{vs}, G with L, $G_p^{0.01}$ with $L_p^{0.01}$, $G_p^{0.05}$ with $L_p^{0.05}$, and $G_p^{0.10}$ with $L_p^{0.10}$. However, the difference between methods L, $L_p^{0.01}$, $L_p^{0.05}$ and $L_p^{0.10}$ and their corresponding Gaussian procedures is negligible.

Estimation of σ^2 should be easier than that of μ, since it is simply a between-replicates variance that does not depend on other parameters. However, σ^2 is hard to estimate using the asymmetric Laplace method, see Figure 4.10. All methods estimate σ^2 equally well, no matter which model generates the data. Comparing the top left and the top right panels of Figure 4.10, we conclude that estimation of σ^2 does not depend on the other parameters.

The parameter q is hard to estimate in all models. In order to get a better estimate of q, the total number of variables V, which in our case is 50, and the number of types T, which is 10, should be large. The quality of estimation depends also on the signal-to-noise ratio $\sigma_\theta^2/\sigma_\eta^2$ for a fixed σ^2. The smaller the signal-to-noise ratio, the more difficult to estimate is q. According to our simulation results, using 10 types and 50 variables may not give precise estimates of q, even when the signal-to-noise ratio is high, see Figure 4.11. For small $\sigma_\theta^2/\sigma_\eta^2$, the hyper-parameter q is computationally unidentifiable. So, we just consider the largest possible signal-to-noise ratio in our simulations, namely $\sigma_\eta^2 = 0.5$ and $\sigma_\theta^2 = 20$ inspired by the metabolite data.

In Figure 4.11, we just consider procedures G_{vs} and L_{vs} because they are the only methods that estimate q. For $p = 0.9$ and $q = 0.1$, the Gaussian procedure, G_{vs}, can estimate q well in all models; see the right panel of Figure 4.11. However, estimating q using the Gaussian fit is biased for data generated from the asymmetric Laplace model; see the green boxplots. It

appears that the Gaussian fits are not reliable for $p = 0.1$ and $q = 0.9$ and considerably under-estimate q; see the left panel of Figure 4.11. However, there is a probability $(1-p)^C$ that an activated variable has no active variable-cluster combination and becomes essentially inactive. This is considerable when p or C is small; see for example the top panels of Figures 4.1 and 4.5, where a lot of variables are activated, but they receive no red blobs in the simulated profiles, meaning no true effects appear for that variable. The probability that a potentially active variable becomes practically active is $1 - (1 - p)^C$; assuming on average that we have 6 clusters and $p = 0.1$, this probability of active variables is 0.42, which agrees with Figure 4.11.

Estimation of the proportion of active variable-cluster combinations for active variables, p, is also difficult. So, we study the effectiveness of estimation when it is easier to estimate p, namely when $\sigma_\eta^2 = 0.5$ and $\sigma_\theta^2 = 20$. The hyper-parameters p and q are inter-related, as having $p = 0$ means q cannot be estimated and vice versa. Like q, estimation of p is biased even for a Gaussian model fitted on Gaussian data, see the white boxplots in the top right panel of Figure 4.12. This is the effect of biased estimation of q which affects p too. Under-estimation of q leads to over-estimation of p, such that pq remains close to the simulated value. This is visible in the bottom panels of Figure 4.12.

Estimation of the experimental error variance σ_η^2 is not very difficult, because there are a lot of inactive variable-cluster combinations. We gain information for estimation of σ_η^2, σ^2, and μ for inactive variable-clusters, and for $\sigma_\eta^2 + \sigma_\theta^2$, σ^2, and μ for active combinations. Hence, we gather direct information for estimation of σ_η^2 for inactive variable-clusters which are many, and indirect information for active ones, which are few. Relatively precise estimation of σ_η^2 is confirmed in Figure 4.13. Comparing the two, we conclude that estimation of σ_η^2 is not sensitive to changes in the other parameters. However, tuning both p and q, introduces estimation bias, see methods $G_p^{0.01}, G_p^{0.05}, G_p^{0.10}, L_p^{0.01}, L_p^{0.05}$, and $L_p^{0.10}$. A method that estimates p and q poorly, like G_{vs}, and procedures that estimate p accurately, like G, are similar in estimation of σ_η^2. Generating data from the asymmetric Laplace model (bottom panels), does not change the scenario.

Estimation of the variance of the true effects σ_θ^2 is rather difficult, since in-

formation about this parameter is available only when active variable-cluster combinations appear, which are rare. It is even more difficult to estimate the left and right tails, because the Laplace model performs inefficiently in estimation of other parameters, like p and q, that are closely related to σ_θ^2. According to Figure 4.14 the Gaussian methods G_{vs} and G estimate σ_θ^2 more efficiently than their corresponding Laplace procedures L_{vs} and L. However, asymmetric Laplace estimates using data generated from the Gaussian effects model have a downward bias, because large true effects, in absolute value, are accounted to come from a heavier tail distribution (Laplace) with smaller variance. The Gaussian fits G and G_{vs} have better performance in the top right panel compared with the top left panel, because $pq = 0.25$ in the top right panel but $pq = 0.09$ in the top left panel, the first gives more information for estimation about hyper-parameter σ_θ^2.

Generally, one could say fixing $q = 1$ does not disturb the estimation (compare G with G_{vs}, and L with L_{vs}), and may help toward more efficient estimation of p and σ_θ^2. On the contrary, fixing p and q together may introduce bias; compare for example methods $G_p^{0.01}, G_p^{0.05}, G_p^{0.10}, L_p^{0.01}, L_p^{0.05}$, and $L_p^{0.10}$ with G and L. For data generated from the asymmetric Laplace model (bottom panels of Figure 4.14) with $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$, asymmetric effects are recognised but estimated with a large uncertainty.

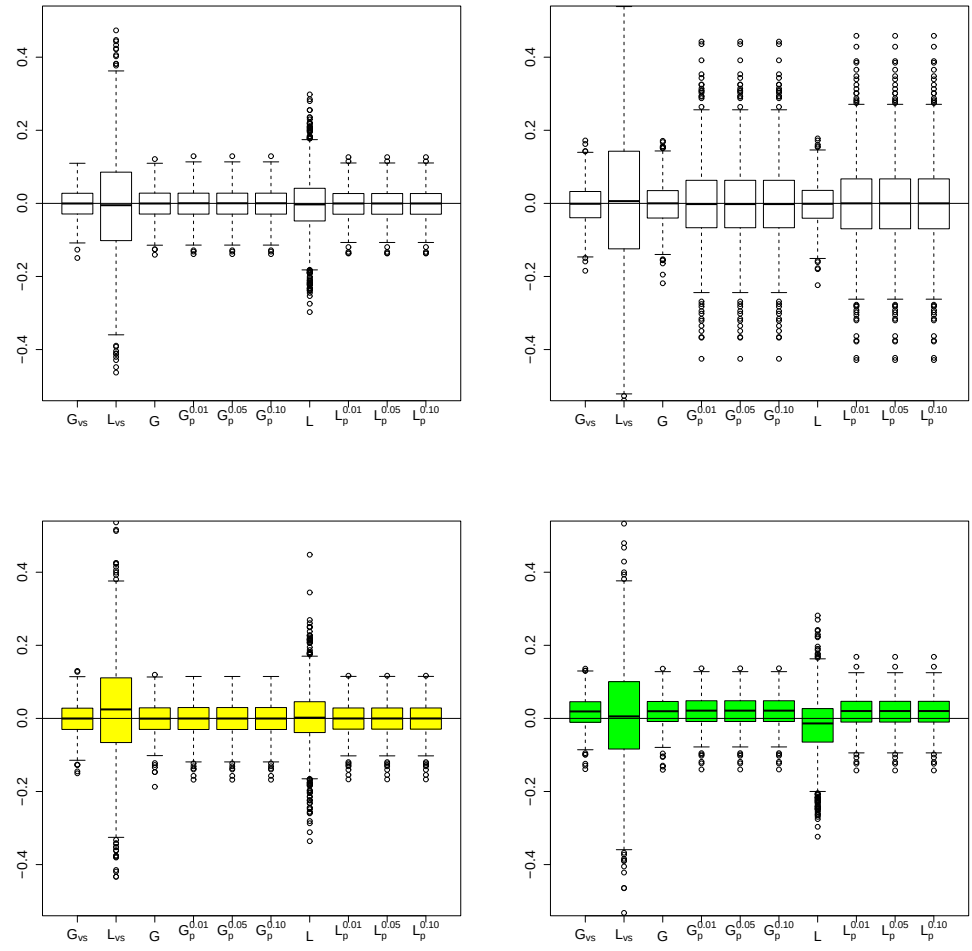

Figure 4.9: Boxplots of $\hat{\mu}$ for data generated with $\mu = 0, \sigma^2 = 1$ and $\sigma_\eta^2 = 0.5$. Top panels with white boxplots correspond to data generated from the Gaussian effects model with $\sigma_\theta^2 = 1, p = 0.1, q = 0.9$ (top left), and $\sigma_\theta^2 = 10, p = q = 0.5$ (top right) panel. Yellow and green boxplots correspond to data generated from the symmetric and the asymmetric Laplace model, respectively. The parameter μ is estimated using the asymmetric Laplace model when data are generated with $\sigma_\theta^2 = 1, \sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 1$ (bottom left) and $\sigma_\theta^2 = 1, \sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$ (bottom right), both with $p = 0.1$ and $q = 0.9$. The true value of μ is zero, represented by the horizontal solid line.

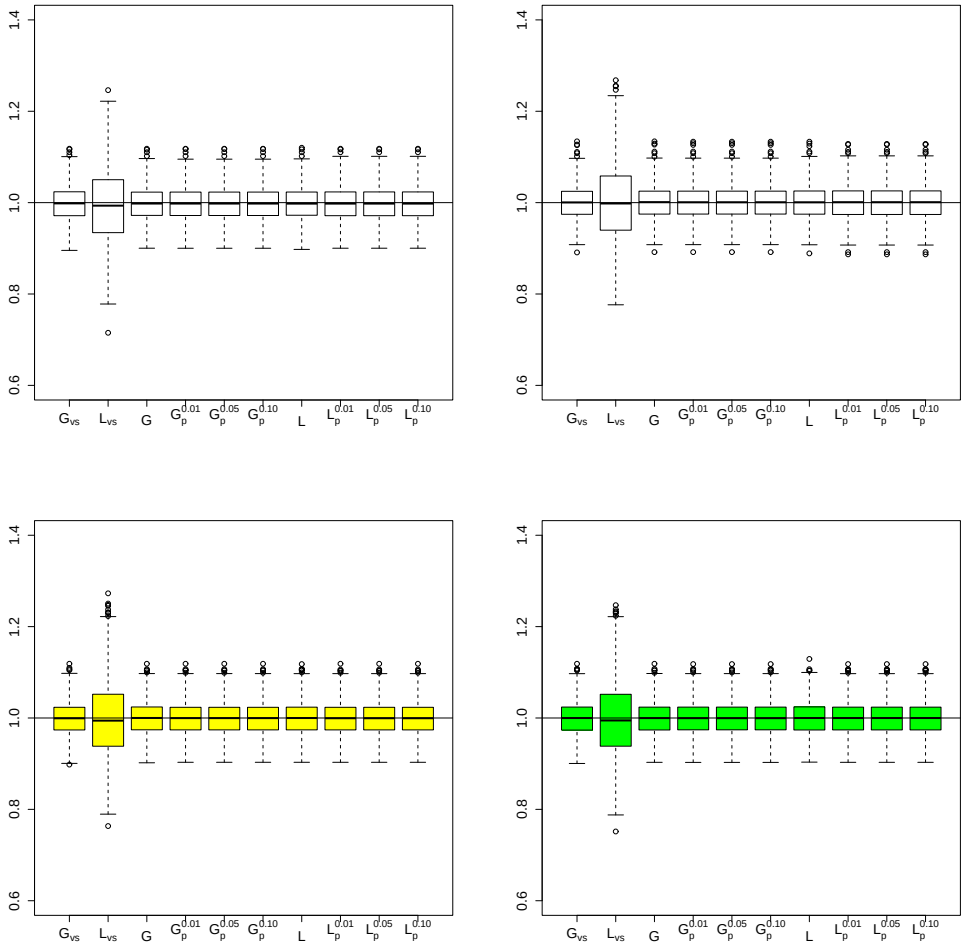

Figure 4.10: Boxplots of $\hat{\sigma}^2$ for data with $\mu = 0, \sigma^2 = 1$ and $\sigma_\eta^2 = 0.5$. White boxplots refer to methods implemented on data sampled from the Gaussian effects model with $\sigma_\theta^2 = 1, p = 0.1, q = 0.9$ (top left), and $\sigma_\theta^2 = 20, p = q = 0.5$, (top right). Yellow boxplots correspond to procedures applied on asymmetric Laplace data with $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 1$ (bottom left), and green ones refer to $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$ (bottom right), both with $p = 0.1$, $q = 0.9$ and $\sigma_\theta^2 = 1$.

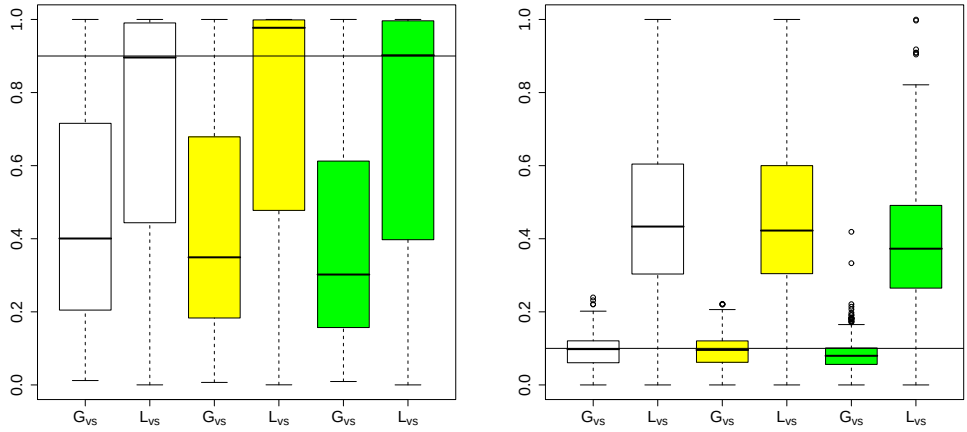

Figure 4.11: Boxplots of of $\hat{q}$ for data generated with $\mu = 0, \sigma^2 = 1, \sigma^2_\eta = 0.5, \sigma^2_\theta = 20$. The procedures implemented on Gaussian effects data are shown using white boxplots, the methods applied to asymmetric Laplace data with $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 1$ are in yellow, and to asymmetric Laplace data with $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 10$ are in green. The left panel refers to data with $p = 0.1, q = 0.9$, and the right panel to $p = 0.9, q = 0.1$.

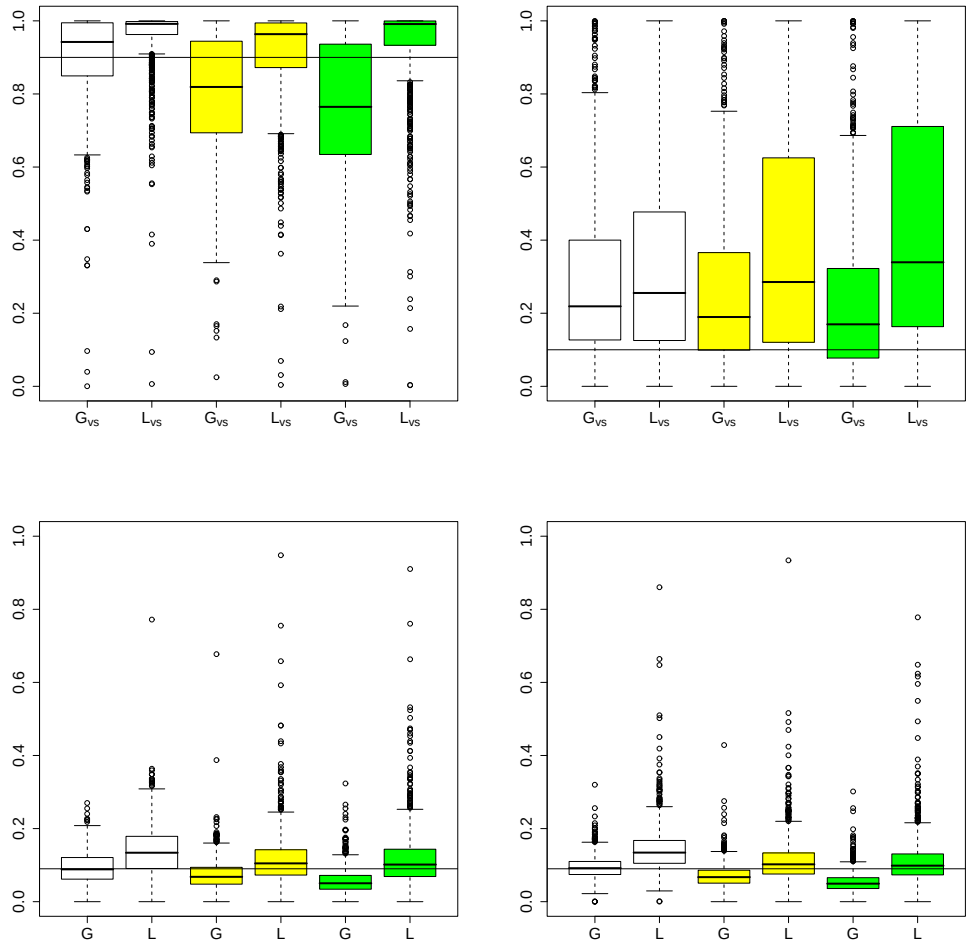

Figure 4.12: Boxplots of $\hat{p}$ for data generated with $\mu = 0, \sigma^2 = 1, \sigma_\eta^2 = 0.5, \sigma_\theta^2 = 20$. In the left panels, p and q are set to 0.9 and 0.1, and in the right panels to 0.1 and 0.9, respectively. The horizontal lines in the top panels are the simulated values of p and in the bottom panels are the simulated values of pq. The white boxplots refer to methods implemented on data generated from the Gaussian effects model, yellow to the symmetric Laplace, and green to the asymmetric Laplace with $\sigma_{\theta_R}^2/\sigma_{\theta_L}^2 = 10$.

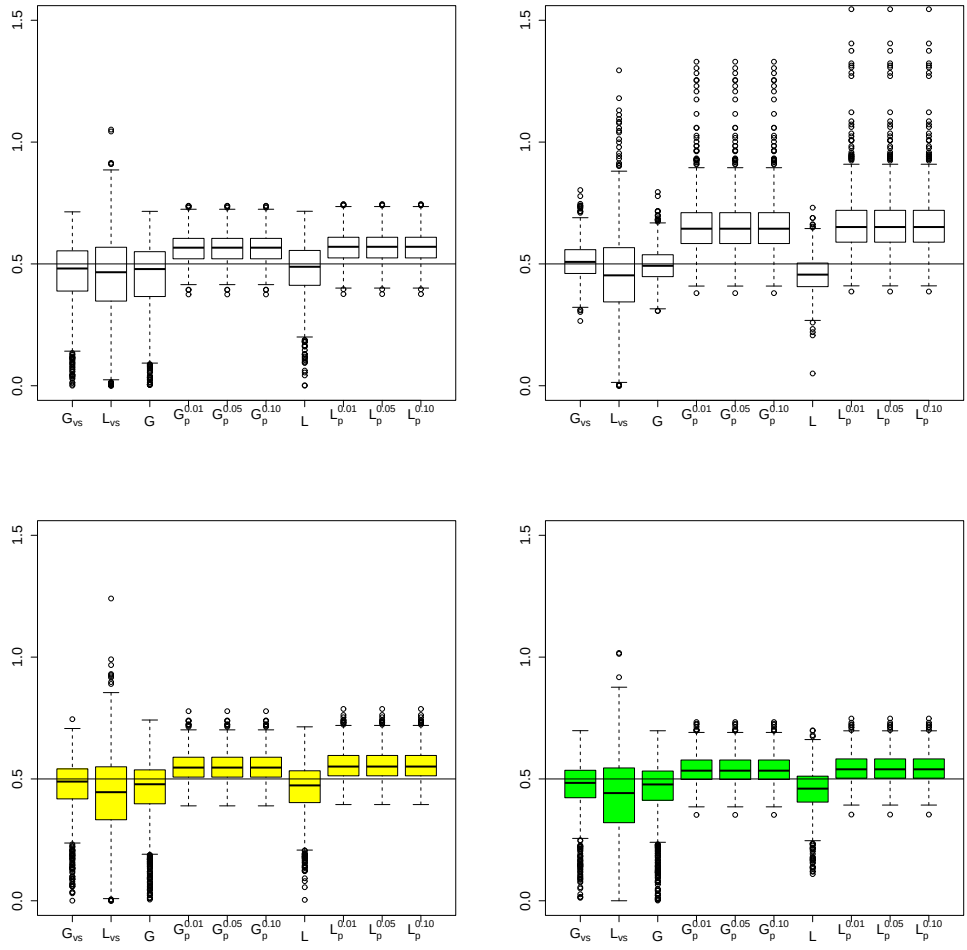

Figure 4.13: Boxplots of $\hat{\sigma}^2_\eta$ for data with $\mu = 0, \sigma^2 = 1, \sigma^2_\eta = 0.5, p = 0.1$, and $q = 0.9$. The top panels are methods applied to Gaussian data with $\sigma^2_\theta = 1$ (top left) and $\sigma^2_\theta = 10$, (top right). The same procedures are implemented on asymmetric Laplace data with $\sigma^2_\theta = 1$, $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 1$ (bottom left) and $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 10$, (bottom right).

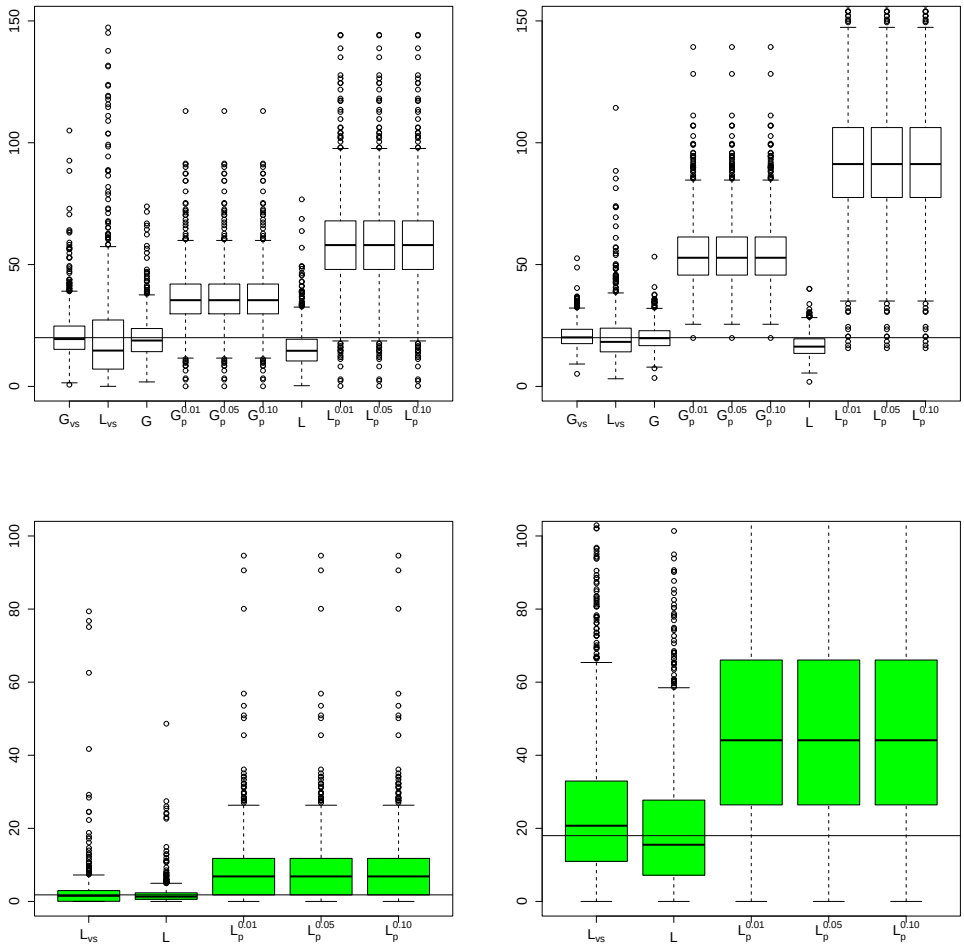

Figure 4.14: Boxplots of $\hat{\sigma}^2_\theta$ for simulated data with $\mu = 0, \sigma^2 = 1, \sigma^2_\eta = 0.5$, and $\sigma^2_\theta = 20$. The top left panel refers to Gaussian data with $p = 0.1, q = 0.9$, the top right with $p = q = 0.5$. The bottom panels are the asymmetric Laplace fits on data generated from the asymmetric Laplace model with $\sigma^2_{\theta_R}/\sigma^2_{\theta_L} = 10$. The bottom left panel corresponds to the estimated values of $\sigma^2_{\theta_L}$ and the bottom right panel to $\sigma^2_{\theta_R}$. The true values are represented by solid horizontal lines.

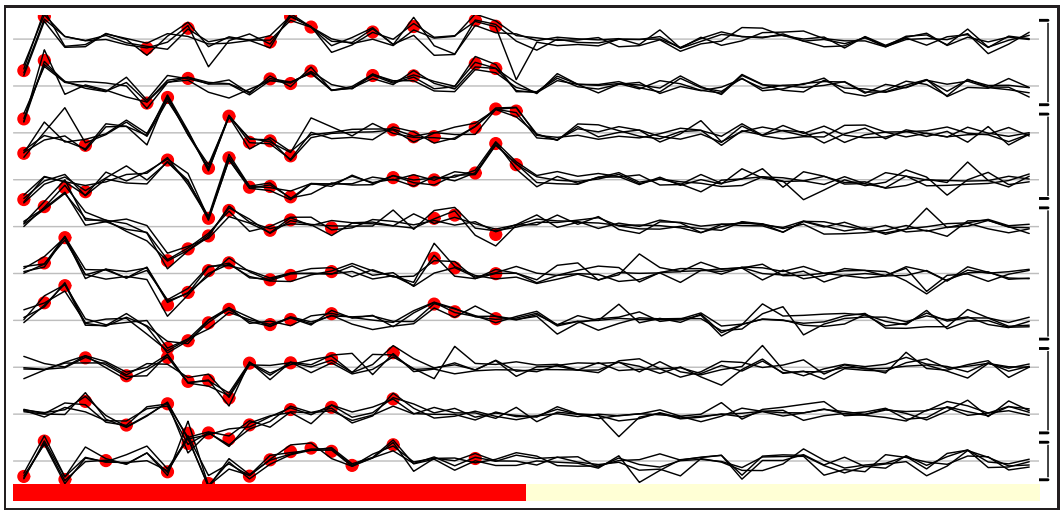

Figure 4.15: Profile plot of simulated data with parameters $\mu = 0, \sigma^2 = 1, \sigma_\eta^2 = 0.5$ $\sigma_\theta^2 = 10$. Experimental error, η_{vct}, and measurement error, ε_{vctr}, are sampled from Student's t distribution with 5 degrees of freedom.

4.5 Heavy-tailed errors

In order to study the efficiency of the clustering procedures in the presence of outliers we apply our clustering methods on data having Gaussian effects but with experimental errors, η_{vct}, and measurement errors, ε_{vctr}, coming from a Student's t distribution with 5 degrees of freedom, t_5. The t_5 distribution is scaled to have variance σ_η^2 for the experimental error layer, and $\sigma^2 = 1$ for measurement error. We consider only $p = q = 0.5$. Hence, the simulation results in Table 4.10 and Figure 4.16 are comparable with these in Table 4.6 and Figure 4.4. The profile plot of data with errors generated from Student's t_5 distribution is given in Figure 4.15.

The effectiveness of clustering procedures for t_5 data, shown in Table 4.10, follows similar pattern as for the Gaussian case in Table 4.6. However, clustering procedures with high signal-to-noise ratios $\sigma_\theta^2/\sigma_\eta^2$ implemented on the heavy-tailed data yields greater losses in terms of both misclassification and trivial losses comparing with the data generated from the Gaussian model. This is due to outliers; however, the losses are not very different. We conclude that our clustering procedures are not very sensitive to heavy-tailed errors. From Figure 4.16, we deduce that our proposed methods are still

significantly preferable to MCLUST applied on principal components.

Loss	Parameter		Fitting Procedure														
	σ_η^2	σ_θ^2	M	G^*	G	G^*_{vs}	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L^*	L	L^*_{vs}	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times$100)	0.5	1	(0.4) 99	(0.7) 94	(0.8) 92	(0.7) 94	(0.9) 91	(0.5) 98	(0.5) 98	(0.5) 97	(0.7) 95	(0.9) 90	(0.7) 95	(1) 89	(0.5) 98	(0.5) 98	(0.5) 97
		10	(1.4) 75	(1.1) 15	(1.4) 25	(1.3) 21	(1.4) 28	(1.6) 49	(1.5) 36	(1.5) 33	(1.1) 15	(1.6) 41	(1.3) 20	(1) 89	(1.6) 52	(1.6) 42	(1.5) 39
		20	(1.5) 66	(0.9) 8	(1) 13	(1.1) 14	(1.2) 17	(1.3) 20	(1) 12	(1) 10	(0.9) 8	(1.3) 20	(1.1) 13	(1) 89	(1.4) 24	(1.2) 17	(1.1) 15
	2	1	(0.1) 100	(0) 100	(0.7) 95	(0) 100	(0.6) 97	(0) 100	(0) 100	(0.1) 100	(0) 100	(0.8) 92	(0.1) 100	(1) 89	(0) 100	(0) 100	(0) 100
		10	(0.7) 95	(1.3) 79	(1.1) 85	(1.3) 80	(1.1) 85	(0.7) 94	(0.8) 92	(0.9) 92	(1.3) 79	(1) 88	(1.3) 80	(1) 89	(0.7) 95	(0.8) 93	(0.8) 92
		20	(1.1) 87	(1.4) 72	(1.2) 81	(1.3) 77	(1.2) 81	(1.2) 82	(1.3) 77	(1.3) 76	(1.4) 71	(1.1) 86	(1.3) 76	(1) 89	(1.2) 84	(1.3) 80	(1.3) 78
Misclassification	0.5	1	(6.5) 360	(7.6) 239	(4.7) 144	(7.6) 233	(4.6) 135	(6.5) 463	(6.8) 410	(6.9) 414	(7.9) 257	(5.7) 163	(7.9) 252	(4.3) 132	(6.4) 486	(6.9) 448	(7.1) 447
		10	(2.3) 60	(0.8) 8	(1.1) 13	(1.2) 13	(1.4) 18	(1.7) 27	(1.3) 17	(1.2) 16	(0.8) 8	(1.6) 26	(1.2) 12	(4.3) 132	(2.5) 38	(2) 26	(2) 25
		20	(1.5) 35	(0.7) 5	(0.7) 6	(0.8) 8	(1) 11	(1) 8	(0.7) 6	(0.8) 6	(0.6) 5	(1) 11	(0.8) 7	(4.3) 132	(1.7) 13	(1.3) 9	(1.1) 9
	2	1	(3.9) 232	(4.6) 579	(5.5) 216	(4.5) 582	(6) 249	(4.5) 495	(4.8) 461	(4.8) 470	(5.3) 555	(6.1) 204	(5.5) 551	(4.3) 132	(4.5) 528	(4.9) 515	(5) 521
		10	(2.7) 98	(2.6) 70	(3.4) 99	(2.7) 78	(3) 90	(6.2) 238	(5) 169	(5) 172	(2.6) 70	(4) 121	(2.7) 76	(4.3) 132	(6.6) 265	(5.7) 197	(5.7) 194
		20	(2.6) 72	(2.5) 62	(3.1) 85	(2.8) 74	(3) 84	(3.7) 97	(2.9) 73	(3) 74	(2.5) 62	(3.6) 106	(2.8) 73	(4.3) 132	(4.2) 113	(3.4) 87	(3.2) 83

Table 4.10: Losses and their standard errors in parentheses above each value. The true effects, θ_{vc}, are generated from the Gaussian distribution with mean 0 and variance σ_θ^2. Experimental error, η_{vct}, and measurement error, ε_{vctr}, are sampled from the Student's t with 5 degrees of freedom with variances σ_η^2 and $\sigma^2 = 1$, respectively. The other parameters are set to $\mu = 0, p = 0.5$, and $q = 0.5$.

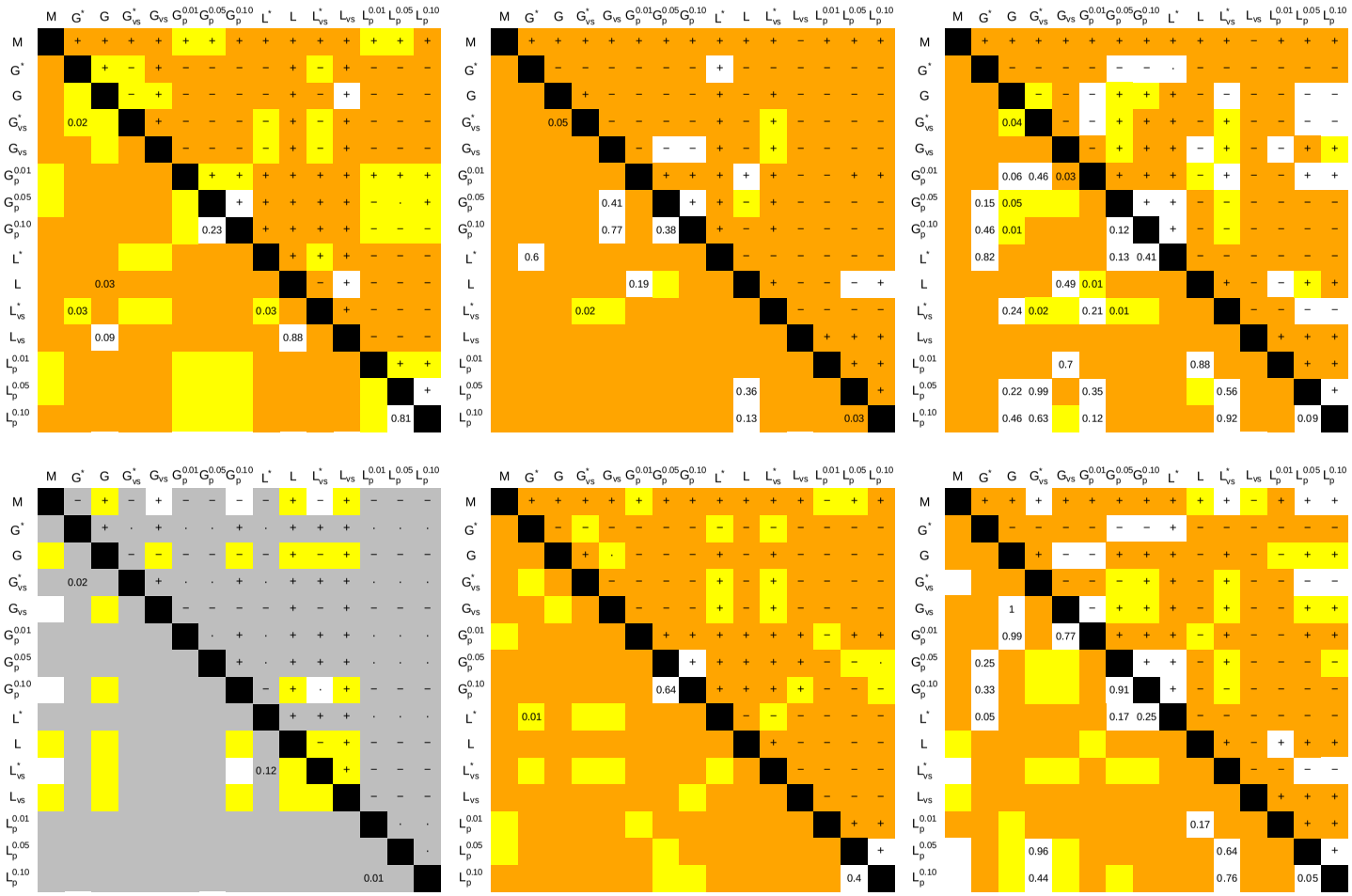

Figure 4.16: Significance tests for data with effects sampled from a Gaussian distribution with mean zero and variance σ_θ^2. Experimental error, η_{vct}, and measurement error, ε_{vctr}, are taken from Student t distribution with 5 degrees of freedom, scaled to have variance σ_η^2 and $\sigma^2 = 1$, respectively. Other parameters are set to $\mu = 0, p = 0.5$, and $q = 0.5$. For more details see the caption to Figure 4.6.

4.6 Correlated Observations

In Sections 4.2, 4.3 and 4.5 we studied the performance of clustering procedures for data generated from the Gaussian effects model, asymmetric Laplace effects model and also when errors are generated from Student's t distribution. In the previous studies we considered independent variables, replicated observations, and the existence of an experimental error hierarchy. In this section we break all these assumptions and sample data from a model that MCLUST is designed to handle. We activate two variables and apply MCLUST twice, once on bivariate data, M^*, and once on data projected using principal components after adding noise variables, M. Projecting data loses clustering information (Chang, 1983), so the amount of information lost by projection can be seen by comparing M with M^*. All of our proposed approaches are applied on noisy data and unreplicated observations, that is $R_{ct} = 1$. Data are generated as follows. Unlike previous sections that randomly sampled the number of clusters, here the number of clusters is fixed to 3, generated on two variables. The first cluster is centred at $\Delta \times (-1, -1)$, the second at $(0, 0)$, and the third at $\Delta \times (1, 1)$. The scalar parameter Δ is a measure of difficulty of clustering, chosen to be 3 or 6 for moderately or completely separable clusters. The observations inside clusters are generated independently with the above mentioned means, unit variance, and correlations equal $(0, 0, 0)$, $(-0.9, 0, 0.9)$, and $(-0.9, -0.9, -0.9)$. Digits inside parenthesis refer to correlation of the first, the second, and the third cluster respectively, see Figure 4.17. Overall 40 observations are generated and distributed in three non-empty clusters according to a uniform multinomial-Dirichlet law. This produces the data to which the MCLUST method is applied, denoted by M^*. Then 48 variables are sampled independently from a standard Gaussian distribution, yielding noisy datasets that other procedures are applied to. In order to see the effect of adding noise variables, an independent simulation is implemented with 98 noise variables.

According to Table 4.11, the MCLUST method implemented on projected data is the worst strategy in all cases using both loss functions except when $\Delta = 3$ and $\rho = (-0.9, -0.9, -0.9)$ when method L is the worst technique, caused by inefficient estimation of parameters. When all correlations equal

zero, the performance of our proposed methods is very close to M^* and sometimes better. The loss values are smaller when the three clusters are generated with equal and strongly negative correlation $(-0.9, -0.9, -0.9)$. The reason is that when all correlations are strongly negative, the clusters are more separated, that is, if one calculates the average Mahalanobis distance between centre of clusters, it is highest when the correlations are all equal -0.9. When data are completely separable, $\Delta = 6$, often our proposed methods are preferable to M and M^*. However, comparing our approaches with M^* is unfair because M^* uses the true active variables which in practice are unknown.

Figure 4.18 shows bar charts for different values of ρ and Δ. The active variables are chosen to be the ones having positive $\log B_v^{10}$ of the Gaussian variable selection model (G_{vs}). Figure 4.18 proposes that the distribution of the estimated number of active variables is right-skewed having a mode equal 2. When $\Delta = 3$, then in about 70% of cases the right number of variables, 2, is reported and changing the correlation structure does not affect the result much. When $\Delta = 6$ in about 90% of cases two clustering variables are reported. Therefore we conclude B_v^{10} is a measure for finding important clustering variables, but tends to over-estimate the number of active variables too.

The simulation results of this section show that our clustering procedures are as efficient as MCLUST, even after adding a lot of noise variables, confirming that our approaches are able to extract useful clustering information which is dense in few variables. We conclude that without knowing the active variables they perform almost as well as MCLUST applied to the true useful variables. Furthermore adding 98 noise variables instead of 48 gives a similar performance; compare Tables 4.11 with 4.12. In addition in both tables, methods $G_p^{0.01}$, $G_p^{0.05}$, $L_p^{0.01}$, and $L_p^{0.05}$ perform relatively well, because the tuned value of p is close to the true value. It is hard to estimate the model parameters, especially p, which plays a crucial role in clustering, when a very large number of noise variables are added, so our approach may give poor results in such situations. Therefore fixing $q = 1$ and tuning p may be effective.

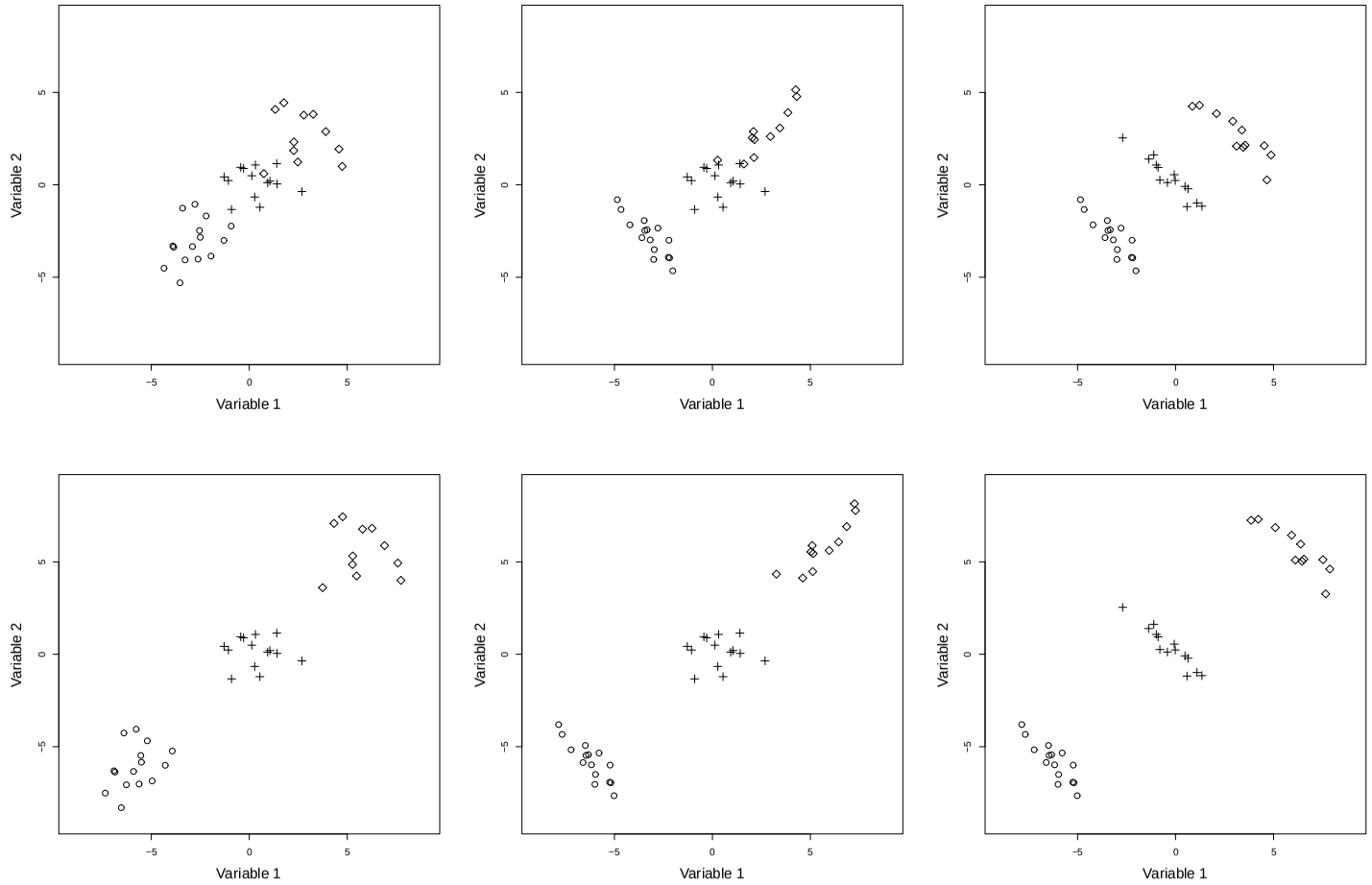

Figure 4.17: Data represented on the two activated variables with three clusters. The numbers of observations in each cluster are distributed with respect to the uniform multinomial-Dirichlet distribution. Subjects are sampled from the bivariate Gaussian distribution with unit variance, means equal to $\Delta \times (-1, -1)$ for the first cluster, $(0, 0)$ for the second cluster and $\Delta \times (1, 1)$ for the third. Correlations are chosen to be $(0, 0, 0)$ (left panels) $(-0.9, 0, 0.9)$ (middle panels) and $(-0.9, -0.9, -0.9)$ (right panels). Each value in the triples corresponds to correlation of one of the three clusters. The scalar parameter Δ equals 3 (top panels) for moderately separable, and equals 6 (bottom panels) for completely separable clusters.

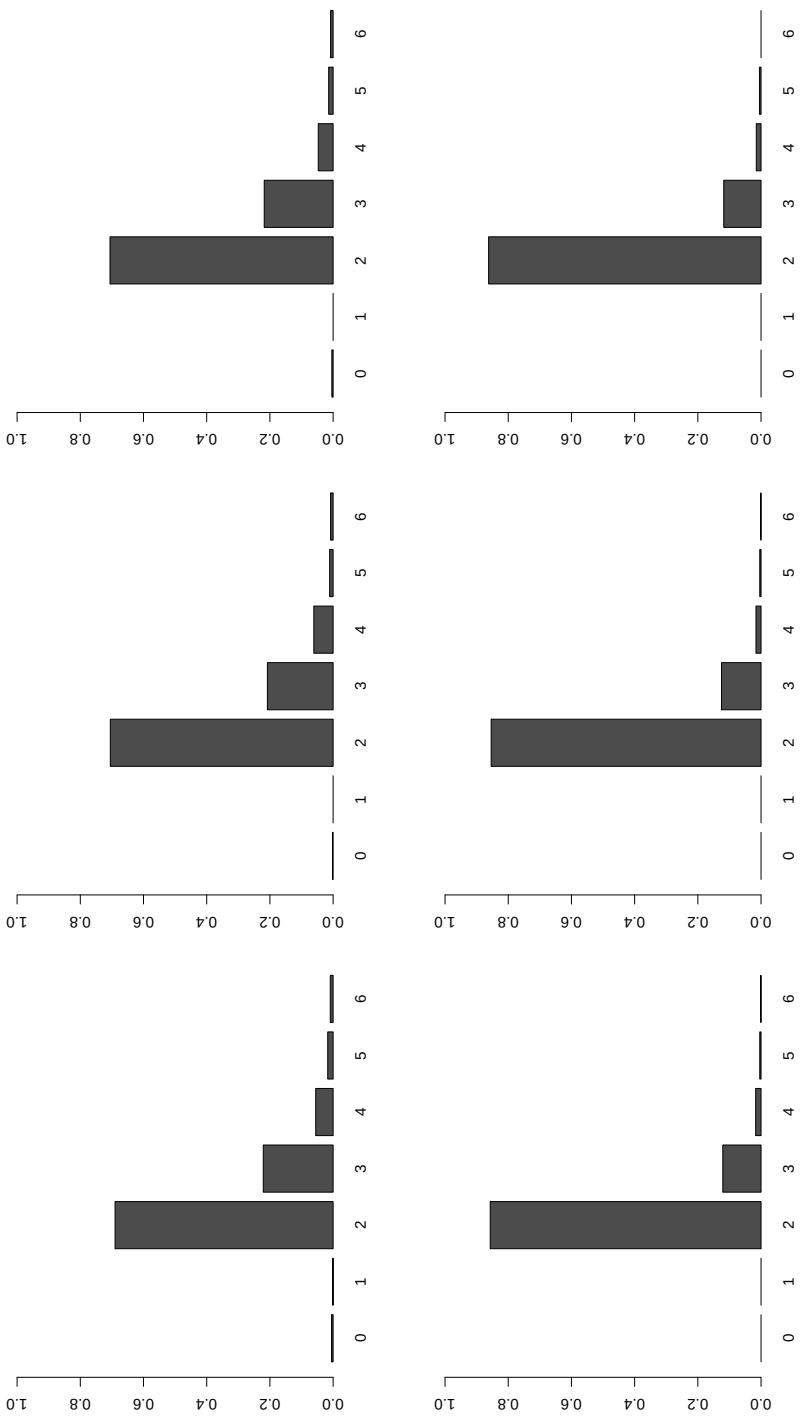

Figure 4.18: Bar charts of the number of variables having $\log B_v^{10} > 0$ using the Gaussian variable selection model (G_{vs}) for data with two clustering variables and 48 noise variables. The top correspond to $\Delta = 3$ and the bottom panels to $\Delta = 6$. The right panels refer to $\rho = (0, 0, 0)$, the middle panels to $\rho = (-0.9, 0, 0.9)$ and the right panels to $\rho = (-0.9, -0.9, -0.9)$. See also the caption to Figure 4.17.

Loss	Δ	Parameter ρ	Fitting Procedure M^*	M	G	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial($\times 100$)	3	$(0,0,0)$	(1.3) 77	(0.7) 94	(1.3) 77	(1.4) 71	(1.3) 78	(1.4) 74	(1.4) 74	(1) 88	(1.4) 72	(1.2) 81	(1.3) 77	(1.3) 76
		$(-0.9,0,0.9)$	(1) 88	(0.6) 96	(1.2) 84	(1.3) 79	(1.2) 83	(1.2) 81	(1.2) 81	(0.9) 91	(1.2) 81	(1.1) 86	(1.2) 84	(1.2) 83
		$(-0.9,-0.9,-0.9)$	(1.4) 25	(1.5) 38	(1.4) 30	(1.4) 26	(1.4) 26	(1.3) 22	(1.3) 22	(1.5) 61	(1.5) 31	(1.5) 30	(1.4) 26	(1.4) 26
	6	$(0,0,0)$	(0.8) 7	(1.3) 22	(0.3) 1	(0.3) 1	(0.2) 0	(0.2) 1	(0.5) 2	(0.4) 1	(0.2) 1	(0.2) 0	(0.2) 1	(0.5) 2
		$(-0.9,0,0.9)$	(1.5) 30	(1.5) 35	(0.5) 3	(0.9) 8	(0.4) 2	(0.5) 2	(0.6) 4	(0.6) 4	(0.8) 7	(0.4) 2	(0.5) 3	(0.7) 5
		$(-0.9,-0.9,-0.9)$	(1) 12	(1.2) 18	(0.4) 2	(1) 12	(0.2) 0	(0.4) 2	(0.6) 4	(0.5) 3	(1) 12	(0.1) 0	(0.4) 2	(0.7) 4
Misclassification	3	$(0,0,0)$	(4.3) 90	(5.6) 209	(1.7) 46	(1.4) 39	(1.6) 49	(1.4) 41	(1.4) 41	(3.2) 89	(1.7) 42	(1.7) 53	(1.5) 46	(1.4) 43
		$(-0.9,0,0.9)$	(2.8) 87	(5) 178	(1.7) 53	(1.5) 47	(1.5) 52	(1.3) 47	(1.4) 48	(3.4) 106	(1.6) 50	(1.6) 57	(1.5) 50	(1.4) 50
		$(-0.9,-0.9,-0.9)$	(2.7) 41	(2.9) 51	(1.1) 13	(0.9) 11	(0.7) 9	(0.6) 8	(0.8) 9	(3.3) 67	(1.2) 15	(0.8) 11	(0.7) 9	(0.8) 10
	6	$(0,0,0)$	(1.2) 8	(2.1) 26	(0.1) 0	(0.1) 0	(0.1) 0	(0.1) 0	(0.2) 1	(0.1) 0	(0) 0	(0.1) 0	(0.2) 0	(0.2) 1
		$(-0.9,0,0.9)$	(1.9) 27	(2.3) 36	(0.2) 1	(0.5) 4	(0.1) 0	(0.2) 1	(0.3) 2	(0.4) 2	(0.5) 3	(0.1) 0	(0.3) 1	(0.3) 2
		$(-0.9,-0.9,-0.9)$	(2.1) 20	(2.5) 31	(0.2) 1	(0.6) 5	(0.1) 0	(0.2) 1	(0.4) 2	(0.3) 1	(0.7) 6	(0) 0	(0.2) 1	(0.4) 2

Table 4.11: Losses and their respective standard errors in parentheses above each value. Two variables are activated and three clusters are built, centred at $\Delta \times (-1,-1)$, $(0,0)$, and $\Delta \times (1,1)$. The number of observations in each cluster is distributed respect to the uniform multinomial-Dirichlet law and sampled from the bivariate Gaussian with mentioned means, unit variances and the correlations ρ. Each value of ρ corresponds to correlation of each cluster. The MCLUST is applied to the bivariate data, denoted by M^*. Then, 48 noise variables are added, independently sampled from a standard Gaussian distribution, and the other procedures are implemented on the noisy data.

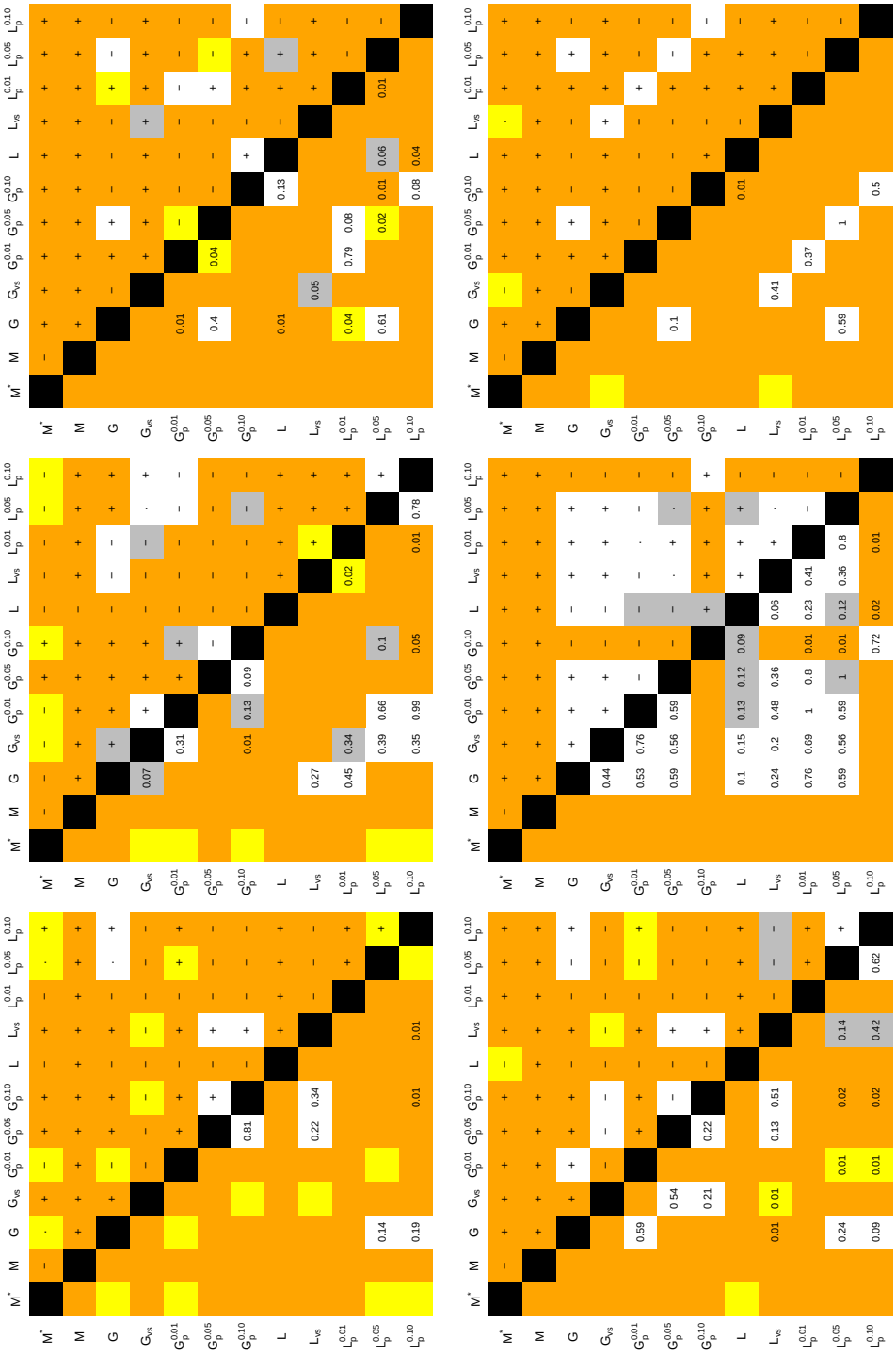

Figure 4.19: Significance tests with two active variables and 48 noise variables corresponding to Table 4.11. For more details see the captions to Figures 4.6 and 4.19.

Loss	Δ	Parameter ρ	Fitting Procedure M^*	M	G	G_{vs}	$G_p^{0.01}$	$G_p^{0.05}$	$G_p^{0.10}$	L	L_{vs}	$L_p^{0.01}$	$L_p^{0.05}$	$L_p^{0.10}$
Trivial(×100)	3	$(0,0,0)$	(1.3) 77	(0.5) 98	(1.3) 78	(1.5) 69	(1.3) 76	(1.4) 75	(1.3) 78	(1.1) 86	(1.4) 72	(1.3) 78	(1.3) 76	(1.3) 77
		$(-0.9,0,0.9)$	(1) 89	(0.5) 97	(1.2) 84	(1.3) 79	(1.2) 81	(1.2) 81	(1.2) 84	(0.9) 90	(1.2) 81	(1.2) 83	(1.2) 82	(1.2) 83
		$(-0.9,-0.9,-0.9)$	(1.3) 23	(1.6) 46	(1.5) 32	(1.4) 28	(1.4) 25	(1.4) 24	(1.4) 29	(1.6) 56	(1.4) 29	(1.4) 28	(1.4) 28	(1.5) 31
	6	$(0,0,0)$	(0.9) 8	(1.4) 24	(0.3) 1	(0.2) 0	(0.2) 0	(0.4) 2	(0.7) 6	(0.4) 2	(0.2) 1	(0.3) 1	(0.4) 1	(0.6) 4
		$(-0.9,0,0.9)$	(1.4) 29	(1.5) 36	(0.5) 2	(0.9) 9	(0.4) 2	(0.5) 3	(0.8) 7	(0.5) 3	(0.8) 8	(0.4) 2	(0.5) 3	(0.7) 6
		$(-0.9,-0.9,-0.9)$	(1) 10	(1.2) 18	(0.4) 2	(1.1) 13	(0.3) 1	(0.5) 2	(0.8) 7	(0.5) 2	(1) 12	(0.3) 1	(0.4) 2	(0.7) 5
Misclassification	3	$(0,0,0)$	(4.3) 91	(5.4) 236	(1.6) 47	(1.4) 38	(1.5) 46	(1.4) 43	(1.5) 46	(2.9) 76	(1.5) 41	(1.6) 50	(1.5) 46	(1.7) 47
		$(-0.9,0,0.9)$	(2.8) 86	(5.1) 205	(1.9) 57	(1.4) 49	(1.5) 52	(1.5) 51	(1.5) 54	(2.8) 84	(1.6) 52	(1.6) 56	(1.6) 54	(1.6) 56
		$(-0.9,-0.9,-0.9)$	(2.5) 34	(4) 78	(1.1) 14	(0.9) 12	(0.7) 9	(0.7) 9	(0.8) 11	(3) 50	(1.3) 15	(0.8) 11	(0.8) 11	(1.2) 14
	6	$(0,0,0)$	(1.3) 9	(2) 26	(0.1) 0	(0) 0	(0.1) 0	(0.1) 0	(0.4) 2	(0.1) 0	(0.1) 0	(0.1) 0	(0.1) 0	(0.2) 1
		$(-0.9,0,0.9)$	(1.9) 25	(2.1) 33	(0.1) 1	(0.5) 4	(0.1) 0	(0.2) 1	(0.4) 2	(0.3) 1	(0.5) 4	(0.1) 0	(0.2) 1	(0.3) 2
		$(-0.9,-0.9,-0.9)$	(1.9) 16	(2.3) 26	(0.2) 1	(0.6) 6	(0.1) 0	(0.2) 1	(0.4) 2	(0.2) 1	(0.6) 6	(0.1) 0	(0.2) 1	(0.3) 2

Table 4.12: Losses and their respective standard errors mentioned in parentheses above each value. Two variables are activated, and 98 noise variables are added. For more details see the caption to Table 4.11.

4.7 Summary

From the simulations discussed in this chapter, we conclude that

- Our Gaussian and asymmetric Laplace procedures are more efficient than MCLUST applied on two principal components.
- They are similar to MCLUST implemented on the true active variables and sometimes better, even after adding a lot of noise.
- The proposed methods are relatively robust to outliers (Table 4.10) and the assumption of independence of variables (Tables 4.11 and 4.12). However, their performance is affected by poor parameter estimation, as often happens for the asymmetric Laplace effects model.
- For fixed model parameters, the Gaussian and asymmetric Laplace clustering are similar, confirming that the mixing distribution is not very important.
- Estimating the proportion of active variables, q, is hard (Figure 4.11), but it does not affect the clustering performance, because fixing q helps a more precise estimation of the remaining parameters. One may fix $q = 1$, then estimate or tune p to a reasonable value and get convincing results. However, the resulting clustering is sensitive to a wrong choice of p (Tables 4.6 and 4.9).
- Situations with the number of noise variables more than 100 are not considered in simulations because it is hard to estimate the model parameters; we do not propose our clustering methods in such cases, unless crucial parameters, such as p, are appropriately tuned.
- Finding important clustering variables using the Bayes factor B_v^{10} is an effective method and is robust with respect to the independence assumption of variables, but may over estimate the number of active variables.

Chapter 5

Conclusion and Discussion

This research shows usefulness of a random effects parametric linear model for clustering high-dimensional observations using a Bayesian approach. The contribution of this work to clustering is the proposed model. However, the linear models especially the fixed effects models are old and well-studied models (Rao, 2001; Graybill, 1976), reabsorbed attention in the recent decades and found to be useful in analysing high-dimensional data (Efron *et al.*, 2004).

Our proposed model can be generalised in various ways, but this may disturb the analytical tractability of the marginal posterior and consequently the implementation of a fast clustering approach may not be easily feasible.

Statistical inference using random effects models (Searle *et al.*, 1992) and mixed effects models (McCulloch and Searle, 2001) also is well-developed, widely discussed and their theory is well-established. It is known that the maximum likelihood estimators, especially the variance components, are sensitive to the assumed mixing distribution (Heckman and Singer, 1984), so Laird (1978) proposed parameter estimation using nonparametric maximum likelihood. In our simulations we found that even if a right distribution is assumed for the mixing components, it is sometimes difficult to get the maximum likelihood estimates for certain mixing distributions. Therefore one direction of continuing this research could be the estimation of the parameters using nonparametric maximum likelihood. However, if the parameters are estimated in a distribution-free manner it is not straightforward to establish a fast clustering method that incorporates variable selection with no

assumption on the distribution of mixing components. This may be regarded as another direction for future developments.

The parameters of our suggested models are estimated using maximum likelihood. However, the estimation may become difficult when a few clustering variables exists in data or distributional assumptions are wrong. In order to help the optimisation routine, one may tune a few of the parameters and estimate the others. Our experience with different datasets shows once reasonable parameters are chosen, the clustering result is convincing.

One way of generalising the variable selection model (2.14) is selecting a group of variables by including another Bernoulli variable. However, we believe it will be more difficult to estimate the model parameters for such models.

Another way of generalising model (2.14) is by selecting variables using the cluster variance as well as the cluster mean, that is assuming a mixture distribution for the measurement error variance or the experimental error variance. In high dimensions often a small subset of variables are useful for clustering and variables that are useless according to the first moment (mean) are rarely useful according the second moment (variance). Furthermore, low sample sizes often do not allow a reliable estimation of covariance matrix even for the effective variables. We believe for high-dimensional data incorporating variance complicates the model and slows down the clustering procedure, but does not improve the clustering result considerably.

The proposed linear model is useful for clustering continuous data and can be generalised for clustering categorised data through the generalised linear models. However, it is not trivial to obtain closed form joint posterior densities for generalised linear models. Therefore, fast clustering is not straightforward and needs more research.

Sometimes genes are transcribed or metabolites are analysed during a specific period, hence time-dependent and high-dimensional data are produced. It would be interesting to investigate if a similar approach can be applied to cluster time series data and choose relevant variables simultaneously.

Stochastic optimisation methods such as Markov chain Monte Carlo are not discussed in this thesis because according to our experience for low sample sizes, dendrograms provide a good approximation to the posterior mode, but

creating an efficient Markov chain to explore the space effectively is not easy. Even if so, stochastic search provides no visual guide to other possible groupings.

Simulations shows that if model parameters are reliably estimated, the parametric distribution of the mixing components has a little effect on clustering result but there is no theoretical argument explaining why this happens. Similar results is reported in Bhowmick *et al.* (2006) in classification. Therefore research on robust model parameters estimation in clustering is demanded. More theoretical studies are also required about sensitivity of the clustering result to a different parametric choice of the mixing distribution.

Clustering is an old data analysis technique but there are few theoretical discussions on it (Hartigan, 1985), maybe because it is hard to study the data grouping as a mathematical object. Model-based clustering by mixture modelling was started few decades ago (McLachlan and Basford, 1988), but statisticians have recently regarded the data grouping as a statistical parameter to be estimated.

We do not have a well-established asymptotic theory for a clustering method. Even if we have such a theory for a particular clustering procedure, often such a theory is useless for high-dimensional-low-sample-size situations due to overfitting; an example is the study of Bickel and Levina (2004) in classification. The asymptotic result must be adjusted for the cases that dimension increases with sample size and this might be regarded as another direction of the future theoretical research in model-based clustering.

Penalisation using the L_1 norm, the lasso of Tibshirani (1996), is found to be useful for high-dimensional regression and classification (Park and Hastie, 2007). High-dimensional clustering using the L_1 penalisation is proposed by Wang and Zhu (2008), but they loose the tree representation of clustering. Their method is not automatic and appropriate choice of the penalising constant is troublesome.

The clustering algorithm provided by this thesis is slow if the number of clustering subjects exceeds 500 subjects, which is rare in metabolomic and gene studies. The algorithm becomes slow when the number of subjects increases because the dissimilarity measure, the marginal posterior, is calculated using the original data. In order to provide a computationally

efficient method one should apply a Lance-Williams type formula (Maechler *et al.*, 2005) for a model-based dissimilarity measure. That is evaluating the dissimilarity measure for the next step of hierarchical clustering using the previous dissimilarity values, preferably a linear combination of the previous values with fixed coefficients. Bayesian models that provide such efficient clustering algorithms have not been discussed.

Ensemble methods such as boosting and bagging have been proposed to aggregate individually weak classifiers in order to obtain a more precise classifier. However, it is not clear how one can implement ensemble methods in clustering because usually there is no information available about misclassified observations after grouping. Research on application of ensemble methods in clustering has been recently started (Domeniconi and Al-Razgan, 2009).

The clustering algorithm proposed in this thesis uses a linear model with disappearing random effect components. Linear random effect models have already been suggested for Bayesian clustering (Heard *et al.*, 2006). However appearance of the random effects in our model is controlled by Bernoulli variables at two levels, the variable-cluster level, and the variable level. The Bernoulli variables can be used to quantify the importance of the variable-cluster and variables after fitting the model. As a consequence of our proposed models, the marginal posterior density is analytically tractable and is a convex combination of two densities, a density that guides the clustering and another which down-weights the effect of useless variables. This is why our clustering method is resistant to noise.

We are not the first to propose introducing Bernoulli variables to implement Bayesian variable selection in clustering. Kim *et al.* (2006) and Tadesse *et al.* (2005) also suggested this method, but the marginal posterior of their models is intractable. Consequently their model parameters cannot be estimated using data. Furthermore, their approach requires reversible jump Markov chain Monte Carlo and hence is slow to fit. The dendrogram representation, which usually practitioners are interested in, is not straightforward either.

The provided methodology in this thesis is automatic, simple, fast, and can sort variables according to their contribution in forming clusters. Our

clustering algorithm has two main advantages. The first is giving an importance measure for variables which can be re-expressed in probability terms. The second is producing dendrograms with probabilistic interpretation. The only competitive clustering method that is fast and gives the variable importances, is the COSA of Friedman and Meulman (2004). However the COSA is not automatic and lacks a probabilistic interpretation for its dendrogram and its variable importances.

The clustering prior used in this thesis prefers small number of clusters and is exchangeable. Booth *et al.* (2008) argue that the prior proposed in McCullagh and Yang (2006) enjoys a sort of consistency in addition to exchangeability and it is feasible to tune a parameter of their prior such that a small number of clusters is preferred a priori. It would be interesting to study which prior works better in practice.

Bibliography

Abonyi, J. and Feil, B. (2007) *Cluster Analysis for Data Mining and System Identification*. Berlin: Birkhauser.

Akaike, H. (1973) Information theory and an extension of the maximum likelihood principle. In *Second International Symposium on Information-Theory*, eds B. N. Petrov and F. Csaki, pp. 267–281.

Alter, O., Brown, P. and Botstein, D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences U.S.A.* **97**, 101–110.

Anderberg, M. R. (1973) *Cluster Analysis for Applications*. London: Academic Press.

Bellman, R. E. (1961) *Adaptive Control Processes: A Guided Tour*. New Jersey: Princeton University Press.

Bensmail, H., Golek, J., Moody, M. M., Semmes, J. O. and Haoudi, A. (2005) A novel approach for clustering proteomics data using Bayesian fast Fourier transform. *Bioinformatics* **21**, 2210–2224.

Bernardo, J. M. and Smith, A. F. M. (1994) *Bayesian Theory*. New York: Wiley.

Bhowmick, D., Davison, A. C., Goldstein, D. R. and Ruffieux, Y. (2006) A Laplace mixture model for identification of differential expression in microarray experiments. *Biostatistics* **7**, 630–641.

Bickel, P. J. and Levina, E. (2004) Some theory for Fisher's discriminant function, naive Bayes, and some alternatives when there are more variables than observations. *Bernoulli* **10**, 989–1010.

Binder, D. A. (1978) Bayesian cluster analysis. *Biometrika* **65**, 31–38.

Binder, D. A. (1981) Approximations to Bayesian clustering rules. *Biometrika* **68**, 275–285.

Bondell, H. D. and Reich, B. J. (2008) Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics* **64**, 115–123.

Booth, J. G., Casella, G. and Hobert, J. P. (2008) Clustering using objective functions and stochastic search. *Journal of the Royal Statistical Society, Series B* **70**, 119–139.

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations (with discussion). *Journal of the Royal Statistical Society, Series B* **26**, 211––252.

Breiman, L. (1996) Bagging predictors. *Machine Learning* **26**, 123–140.

Brown, P. J., Vannucci, M. and Fearn, T. (1998) Multivariate Bayesian variable selection and prediction. *Journal of the Royal Statistical Society, Series B* **60**, 627–641.

Candes, E. J. and Tao, T. (2007) The Dantzig selector: Statistical estimation when p is much larger than n (with discussion). *Annals of Statistics* **35**, 2313–2404.

Chan, Y.-B. and Hall, P. (2009) Scale adjustments for classifiers in high-dimensional, low sample size settings. *Biometrika* **96**, 469–478. to appear in Biometrika.

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

Chaudhary, A. (2007) An introduction to the interface between C and R. Technical report, Institute of Mathematics, Ecole Polytechnique Fédérale de Lausanne.

Chen, T., Morris, J. and Martin, E. (2006) Probability density estimation via an infinite Gaussian mixture model: application to statistical process monitoring. *Applied Statistics* **55**, 699–715.

Crowley, E. M. (1995) Product partition models for normal means. *Journal of the American Statistical Association* **92**, 192–198.

Dahl, D. B. (2003) An improved merge-split sampler for conjugate Dirichlet process mixture models. Technical report, University of Wisconsin, Madison.

Dasgupta, A. and Raftery, A. E. (1998) Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association* **93**, 294–302.

Davison, A. C. (2003) *Statistical Models*. Cambridge: Cambridge University Press.

Dempster, A., Laird, N. and Rubin, D. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B* **39**, 1–38.

Diaconis, P. and Friedman, J. H. (1984) Asymptotics of graphical projection pursuit. *Annals of Statistics* **12**, 793–815.

Domeniconi, C. and Al-Razgan, M. (2009) Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data* **2**, 1–40.

Efron, B., Hastie, T. J., Johnstone, I. M. and Tibshirani, R. J. (2004) Least angle regression (with discussion). *Annals of Statistics* **32**, 407–499.

Everitt, B., Landau, S. and Leese, M. (2001) *Cluster Analysis*. London: Edward Arnold.

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**, 179–188.

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

Freije, W. A., Castro-Vargas, F. E., Fang, Z., Horvath, S., Cloughesy, T., Liau, L. M., Mischel, P. S. and Nelson, S. F. (2004) Gene expression profiling of Gliomas strongly predicts survival. *Cancer Research* **64**, 6503–6510.

Freund, Y. (1995) Boosting a weak learning algorithm by majority. *Information and Computation* **121**, 256–285.

Friedman, J. H. (1987) Exploratory projection pursuit. *Journal of the American Statistical Association* **82**, 249–266.

Friedman, J. H. (1989) Regularized discriminant analysis. *Journal of the American Statistical Association* **84**, 165–175.

Friedman, J. H., Hastie, T. J. and Tibshirani, R. J. (2000) Additive logistic regression: a statistical view of boosting. *Annals of Statistics* **28**, 2000.

Friedman, J. H. and Meulman, J. J. (2004) Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society, series B* **66**, 815–849.

Friedman, J. H. and Tukey, J. W. (1974) A projection pursuit algorithm for exploratory data analysis". *IEEE Transactions on Computers, C* **23**, 881—-890.

George, E. I. and McCulloch, R. E. (1997) Approaches for Bayesian variable selection. *Statistica Sinica* **7**, 339–373.

Gohlke, R. S. (1959) Time-of-flight mass spectrometry and gas-liquid partition chromatography. *Analytical Chemistry* **31**, 535–541.

Gohlke, R. S. and McLafferty, F. W. (1993) Early gas chromatography/mass spectrometry. *Journal of the American Society for Mass Spectrometry* **4**, 367–371.

Gordon, A. D. (1999) *Classification*. London: CRC Press.

Graybill, F. A. (1976) *Theory and Application of the Linear Model*. Massachusetts: Wadsworth.

Green, P. J. (1995) Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711–732.

Hand, D. J. (2006) Classifier technology and the illusion of progress. *Statistical Science* **21**, 1–14.

Hand, D. J. and Yu, K. (2001) Idiot's Bayes — not so stupid after all? *International Statistical Review* **69**, 385–399.

Hartigan, J. A. (1985) Statistical theory in clustering. *Journal of Classification* **2**, 63–76.

Hastie, T. J., Tibshirani, R. J. and Buja, A. (1995) Flexible discriminant and mixture models. In *Neural Networks and Statistics*, eds J. W. Kay and D. M. Titterington. New York: Oxford University Press.

Hastie, T. J., Tibshirani, R. J. and Friedman, J. H. (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

Heard, N. A., Holmes, C. C. and Stephens, D. A. (2006) A quantitative study of gene regulation involved in the immune response of *Anopheline* mosquitoes: An application of Bayesian hierarchical clustering of curves. *Journal of the American Statistical Association* **101**, 18–29.

Heckman, J. and Singer, B. (1984) A method for minimizing the impact of distributional assumptions in econometric models for duration data. *Econometrica* **52**, 271–320.

Heller, K. A. and Ghahramani, Z. (2005) Bayesian hierarchical clustering. In *Twenty-second International Conference on Machine Learning*.

Hoff, P. D. (2006) Model-based subspace clustering. *Bayesian Analysis* **1**, 321–344.

Huber, P. J. (1985) Projection pursuit (with discussion). *Annals of Statistics* **13**, 435–525.

Jain, A. K. and Dubes, R. K. (1988) *Algorithms for Clustering Data*. London: Prentice-Hall.

Jain, S. and Neal, R. M. (2004) A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics* **13**, 158–82.

Jain, S. and Neal, R. M. (2007) Splitting and merging components of a nonconjugate Dirichlet process mixture model. *Bayesian Analysis* **2**, 445–472.

James, G. M., Radchenko, P. and Lv, J. (2009) DASSO: connections between the Dantzig selector and lasso. *Journal of the Royal Statistical Society, Series B* **71**, 127–142.

Johnson, R. A. and Wichern, D. W. (2007) *Applied Multivariate Statistical Analysis*. 6th edition. London: Prentice-Hall.

Jones, M. C. and Sibson, R. (1987) What is projection pursuit? (with discussion). *Journal of the Royal Statistical Society, Series A* **150**, 1–37.

Kass, A. E. and Raftery, A. E. (1995) Bayes factors. *Journal of the American Statistical Association* **90**, 773–795.

Kaufman, L. and Rousseeuw, P. J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

Kim, S., Tadesse, M. G. and Vannucci, M. (2006) Variable selection in clustering via Dirichlet process mixture models. *Biometrika* **93**, 877–893.

Laird, N. (1978) Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association* **73**, 805–811.

Lau, J. W. and Green, P. J. (2007) Bayesian model-based clustering procedures. *Compuational Statistics and Data Analysis* **16**, 526–558.

Li, L. and Li, H. (2004) Dimension reduction methods for microarrays with application to censored survival data. *Bioinformatics* **20**, 3406–3412.

Liu, J. S., Zhang, J. L., Palumbo, M. J. and Lawrence, C. E. (2003) Bayesian clustering with variable and transformation selections. In *Bayesian Statistics 7*, eds J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith and M. West, pp. 249–275. New York: Oxford University Press.

Maechler, M., Rousseeuw, P., Struyf, A. and Huber, M. (2005) Cluster analysis basics and extensions. R package cluster: http://cran.r-project.org/web/packages/cluster.

McCullagh, P. and Yang, J. (2006) Stochastic classification models. In *Proceedings of International Congress of Mathematicians*, volume 3. European Mathematical Society.

McCulloch, C. E. and Searle, S. R. (2001) *Generalized, Linear, and Mixed Models*. New York: Wiley.

McLachlan, G. J. and Basford, K. E. (1988) *Mixture Models: Inference and Applications to Clustering*. New York: Marcel Dekker.

McLachlan, G. J., Bean, R. W. and Peel, D. (2002) A mixture model-based aproach to the clustering of microarray expression data. *Bioinformatics* **18**, 413–422.

Medvedovic, M. (2000) Clustering multinomial observations via finite and infinite mixtures and MCMC algorithms. In *Proceedings of the Joint Statistical Meeting: Statistical Computing Section*, pp. 48–51. Indianapolis: American Statistical Association.

Medvedovic, M., Yeung, K. Y. and Bumgarner, R. E. (2004) Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics* **20**, 1222–1232.

Meila, M. (2005) Comparing clusterings — an axiomatic view. http://www.stat.washington.edu/mmp/Papers/icml05-compare-axioms.pdf.

Messerli, G., Partovi Nia, V., Trevisan, M., Kolbe, A., Schauer, N., Geigenberger, P., Chen, J., Davison, A. C., Fernie, A. R. and Zeeman, S. C. (2007) Rapid classification of phenotypic mutants of Arabidopsis via metabolite fingerprinting. *Plant Physiology* **143**, 1481–1492.

Messerli, G. L. Y. (2007) *Starch Degradation in Arabidopsis Thaliana leaves*. Ph.D. thesis, ETH Zurich.

Mitchell, T. J. and Beauchamp, J. J. (1988) Bayesian variable selection in linear regression (with discussion). *Journal of the American Statistical Association* **83**, 1023–1036.

Park, M. Y. and Hastie, T. (2007) L_1 regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society, Series B* **69**, 659—-677.

Raftery, A. E. and Dean, N. (2006) Variable selection for model-based clustering. *Journal of the American Statistical Association* **101**, 168–178.

Rand, W. M. (1971) Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66**, 846–850.

Rao, C. R. (2001) *Linear Statistical Inference and its Applications*. Second edition. New York: Wiley.

Rencher, A. C. (1998) *Multivariate Statistical Inference and Applications*. New York: Wiley.

Rosenblatt, F. (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* **65**, 386–408.

Sand, P. and Moore, A. W. (2001) Repairing faulty mixture models using density estimation. In *Proceedings of the 18th International Conferece on Machine Learning*, pp. 457–464. San Francisco: Morgan Kaufmann.

Schwarz, G. E. (1978) Estimating the dimension of a model. *Annals of Statistics* **6**, 461–464.

Searle, S. R., Casella, G. and McCulloch, C. E. (1992) *Variance Components*. New York: Wiley.

Sokal, R. R. and Sneath, P. H. (1963) *Principles of Numerical Taxonomy*. London: Freeman.

Tadesse, M. G., Sha, N. and Vannucci, M. (2005) Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association* **100**, 602–617.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* **58**, 267–288.

Tryon, R. C. (1939) *Cluster Analysis*. Ann Arbor: Edwards Brothers.

Tryon, R. C. and Bailey, D. E. (1970) *Cluster Analysis*. London: McGraw-Hill.

Vapnik, V. (1996) *The Nature of Statistical Learning Theory*. New York: Springer.

Wang, S. and Zhu, J. (2008) Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics* **64**, 440–448.

Yeung, K. Y., Fraley, C., Murua, A., Raftery, A. E. and Ruzzo, W. L. (2001) Model-based clustering and data transformations for gene expression data. *Bioinformatics* **17**, 977–987.

Vahid PARTOVI NIA Address: FSB, IMA, STAT, Station 8, EPFL, CH-1015, Ecublens, Switzerland. Phone: +41 21 693 5503 E-mail: (vahid.partovinia at epfl.ch), Personal Website: http://probstat.ch/, Sex: Male, Nationality: Iranian, Canadian Immigrant, Birth Date: 1980, Martial Status: Single.

EDUCATION

Ph.D. Ecole Polytechnique Fédérale de Lausanne, Switzerland. MSc. in Mathematical Statistics, Ferdowsi University of Mashhad, Iran, 2004. B.Sc. in Statistics, Ferdowsi University of Mashhad, Iran, 2002.

JOB POSITIONS

Research Fellow at McGill University. Teaching Assistant at Ecole Polytechnique Fédérale de Lausanne. Researcher of the National Centre of Competence in Research in Plant Survival (NCCR), the University of Neuchatel. Data Analyst, IT Organization of The Mashhad Municipality. Statistics Consultant, Ferdowsi University of Mashhad. Experiment Designer-Analyst, Talayeh-Gostaran Quality Company.

TEACHING ASSISTANTSHIPS

Statistics for mathematicians 2 semesters. Statistics for engineers 5 semesters. Categorical Data Analysis 1 semester. Regression 2 semesters. Statistical Methods 1 semester. Multivariate Analysis 1 semester. Time Series Analysis 1 semester. Mathematical Analysis 1 semester.

PROJECTS SUPERVISED

"Survival and Censored Data" by Lafteris Samatzis. "Tree Representation of Monte Carlo Clustering" by Arpid Chaudhary.

HONORS and AWARDS

Top Ranks in National MSc Enterance Exam Among Nearly 2000 Students of Statistics: 2nd Rank in Mathematical Statistics, 5th Rank in Biostatistics, 8th Rank in Applied Statistics. Winner of the outstanding research proposal for master dissertation, Ministry of Science, Research and Technology of Iran. Winner of the student paper award for “Art of Modelling in Statistics” in Farsi. Two-year joint fellowship of the Swiss National Science Foundation (with the McGill university and the Oxford university) for High-Dimensional Bayesian Clustering research proposal.

LANGUAGE ABILITIES

Fluent in Farsi, English and Esperanto, Moderate in French and Arabic.

REFERENCES

Anthony C. Davison (anthony.davison at epfl.ch), Stephan Morgenthaler (stephan.morgenthaler at epfl.ch) , Ali Reza Fotouhi (ali.fotouhi at ucvf.ca).

PUBLICATIONS

Refereed articles

- Parchami, A., Mashinchi, M., and Partovi Nia, V. (2008) *A Consistent Confidence Interval for Fuzzy Capability Index*, **Applied and Computational Mathematics**, Vol. 7, no. 1, 119-125.
- Messerli, G., Partovi Nia, V., Trevisan, M., Kolbe, A., Schauer, N., Geigenberger, P., Chen, J., Davison, A. C., Fernie, A. R. and Zeeman, S. C. (2007) *Rapid Classification of Phenotypic Mutants of Arabidopsis Via Metabolite Fingerprinting*, **Plant Physiology**, Vol. 143, 1484-1492.
- Partovi Nia, V. (2006) *Gauss-Hermite Quadratures: Numerical or Statistical Method?* **The 8th. Iranian Statistical Conference Proceedings** (invited and refereed papers), 209-215.

Working Papers

- Partovi Nia, V. and Davison, A. C. *Fast High-Dimensional Bayesian Classification and Clustering,* submitted.
- Partovi Nia,V. *Label to Dendrogram Package,* submitted.

Contributed conference presentations

- Partovi Nia, V. and Davison, A. C. (2008) *Rapid Variable Selection for Bayesian Clustering* presented in EPFL research Day, poster.
- Davison, A. C. and Partovi Nia, V. (2007) *Fast High-Dimensional Classification and Clustering* presented in Statistical Methods in Bioinformatics in Munich, poster.
- Partovi Nia, V. and Davison, A. C. (2006) *Bayesian Metabolite Fingerprinting* presented in International Biometric Conference in Montreal, poster.

Software and Packages

- labeltodendro: R-Package with Arpit Chaudhary and Anthony C. Davison.
- bclust: R-Package with Anthony C. Davison.