Statistical Methodology Motivated by Problems in Genetics

Matthew Sperrin, MMorse

Department of Mathematics and Statistics

Lancaster University

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

May 2010

Abstract

Sequencing the human genome has made vast amounts of potentially useful genetic data accessible. An important challenge in statistics is to develop methodology to extract information from this data. In this thesis, developments are made in two methodological areas that have wide applications in genetics.

First, probabilistic methods to deal with the label switching problem in Bayesian mixture models are introduced. Mixture models are used in situations where populations may consist of a number of sub-populations, or as a semi-parametric modelling tool. The label switching problem can prevent meaningful interpretation of the output of Markov Chain Monte Carlo samplers. Specifically, inference on attributes specific to sub-populations can be difficult. Such attributes play an important role in understanding genetic effects. We introduce probabilistic relabelling strategies as a natural way of overcoming the label switching problem, and compare with existing strategies. The comparisons demonstrate that the advantages offered by probabilistic strategies come without loss in parameter estimation ability.

Second, we introduce direct effect testing (DET), which is a novel method that distinguishes direct from indirect effects between binary predictors and a binary response. DET consists of two stages: the first stage finds effects, the second stage infers the uncertainty in determining which predictors cause which effects. The method is useful when it is of interest to recover direct effects between a large number of predictors and the response. This is a common goal in genetics, where we are interested in the effects of variations in the genome on the prevalence of a phenotype. This work includes detailed simulations, comparing the ability of a number of methods at recovering direct effects. DET outperforms existing methods at recovering direct effects in situations where there is high correlation between predictors, and matches their performance when the correlation is moderate or small.

Acknowledgements

I begin by thanking Dr Thomas Jaki (Jack), for the excellent supervision, support and advice he has offered me throughout the last two years of my PhD. Earlier in my PhD I was supervised by Prof. Ernst Wit, before his career took him to the Netherlands. The differing styles of supervision of Ernst and Jack have fitted perfectly with my own development as a researcher, and I feel I have been fortunate to have had the opportunity to work with them both.

Furthermore I would like to thank Prof. Paul Fearnhead for expert advice, and to academics at the University of Warwick, particularly Jim Smith, Barbel Finkenstadt and Jane Hutton, who encouraged me to continue with my studies.

The community of young researchers at Lancaster University is excellent, and I would like to extend my thanks to my fellow PhD students for their advice, friendship and support.

I acknowledge EPSRC for supporting me financially through the PhD, and the support staff in the Maths and Statistics department for all their help and gossip!

Finally I would like to thank my family, for allowing me to choose my own path and giving me their unconditional support.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for the award of a higher degree elsewhere.

Matthew Sperrin

List of Papers

This thesis includes the following three papers:

- Sperrin M., Jaki T., Wit E. (2010), *Probabilistic relabelling strategies for the label switching problem in Bayesian mixture models*, Statistics and Computing, Online ahead of print, DOI: 10.1007/s11222-009-9129-8.

- Sperrin M., Jaki T. (2010), *Direct effect testing: A two-stage procedure to test for effect size and variable importance for correlated binary predictors and a binary response*, Submitted.

- Sperrin M., Jaki T. (2010), *Recovering direct effects in genetics: A comparison*, In preparation.

The papers are given as Chapters **??**, **??** and **??** respectively. The relevant appendices can be found at the end of each Chapter, while the bibliography is listed at the end of the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Recent Advances in Genetics

The study of genetics has entered a new era. We have progressed over the last decade from knowing the expressions and chemical make-up of a few genes, to sequencing the entire human genome (**?**). The human genome is mostly homogeneous, with over 99% of its make-up identical for all humans. The interest lies in the remaining 1% or so of the genome that is different from one person to another, as this contributes to human diversity. Diversity of the human species is in itself an interesting and worthwhile topic of study, but for obvious reasons the real interest lies in understanding why some people are more susceptible to certain diseases than others. Whilst it is well known that environmental factors are important (for example, 'smoking causes lung cancer' **?**), here we are interested in how disease susceptibility can be affected by genetic differences. There is a very large number of these genetic differences, hence the Aristotelian school of thought, advocating the understanding of underlying scientific processes, is problematic to apply. The answer lies with evidence based learning and statistics, but that is not to say that statistics has all the answers. The pace of advance in genetics has been so great that statisticians are still trying to catch up in providing methods to understand the vast quantities of data that can now be collected.

All of this means that the development of statistical methods to interpret genetic data is a rich growth area in current and future scientific research. Two particular areas are addressed in this thesis. First, progress is made in dealing with the population structure, or relatedness, between participants in a genetic study, through developments in the theory of mixture models. Second, methods that handle situations where the number of predictors greatly exceeds the number of observations are developed, through the new method of direct effect testing.

1.2 Genetics and Single Nucleotide Polymorphisms

This section gives a very brief outline of the genetics needed to understand this thesis. It is not intended to be complete, see Chapter 2 of **?** for a more detailed account.

The human genome is built out of a four letter alphabet. Each letter corresponds to a *base*, also known as a *nucleotide*. Sequences of these bases contain genetic information. The four letters used in the alphabet are abbreviations of the chemical names of the bases; the names are adenine (represented by A), guanine (G), cytosine (C) and thymine (T). The human genome has 23 pairs of linear sequences of bases called *chromosomes*. Of these, 22 are called *autosomes*, and the remaining one is a sex chromosome. Except for reproductive cells, all cells in the human body contain 46 chromosomes, consisting of the 22 pairs of autosomes and a pair of sex chromosomes. Sex chromosomes are of two types, X and Y. Females have a pair of X chromosomes and males have one X and one Y chromosome. A section of a chromosome may be represented as a string of letters, for example,

$$\ldots \text{AAGTTGCAAATGTTAGT} \ldots$$

There is some diversity between chromosomes, which is the source of genetic differences between humans. If a particular chromosome of one human is compared with the same chromosome of another human, they will be almost identical. Differences occur when there is more than one possible base at a particular location on the chromosome; such locations are called *single nucleotide polymorphisms (SNPs).* Suppose the following two sequences are small sections of the chromosomes of two individuals:

$$\ldots\text{AAGTTGCA}\boldsymbol{A}\text{ATGTTAGT}\ldots$$

$$\ldots\text{AAGTTGCA}\boldsymbol{T}\text{ATGTTAGT}\ldots$$

The position highlighted in both sequences is a SNP, because different bases are found in each individual. Usually, the more common base is seen as the standard one (called the wild type), and the less common base is considered a mutation. Indeed, since chromosomes occur in pairs, these chromosomes could also have been drawn from the same individual. This means that the mutation can occur on one of the chromosomes in the pair (a single mutation) or both chromosomes in a pair (a double mutation).

Following completion of the human genome sequencing in 2001, the HapMap project has set out to identify SNPs, and to date approximately 3 million have been found (**?**). Recent work has focussed on associations between these SNPs and disease; the fundamental question is, do certain SNPs reduce or increase the risk of a certain disease? **?** have recently carried out a large study that has identified many SNPs that are associated with specific diseases. In different sub-populations, different SNPs may be responsible for the same disease. Therefore a study must either take participants from only one sub-population (for example, **?**), or take account of the sub-population structure using a method such as mixture modelling (for example, **?**).

Finally for genetic background, we discuss *linkage disequilibrium*, which is formally defined as a population association between the bases at two locations (**?**). It refers to the correlated behaviour of SNPs. Typically, these correlations are spatial, as SNPs that are nearby on the genome are often closely related, but there can also be long range correlations. Consider two nearby SNPs, S_1 and S_2 say. Suppose that SNP S_1 can use the bases A and T, and SNP S_2 can use the bases C and G. If SNP S_1 takes base A, then SNP S_2 usually takes base C; if SNP S_1 takes base T, then SNP S_2 usually takes base G. Hence, SNPs S_1 and S_2 are correlated, and we say they are in linkage disequilibrium. This correlation between the two SNPs has a major advantage but also a major disadvantage. The advantage is that redundancy in the information carried by SNPs S_1 and S_2 means that only one of them needs to be measured when collecting a genetic sample. The disadvantage is that if an effect were found on SNP S_1, say, it would be very difficult to tell whether the truly influential SNP really is SNP S_1, or is actually SNP S_2. The occurrence of linkage disequilibrium is explained by the way in which organisms reproduce (see **?**).

1.3 Focus of Thesis

This work focuses on the development of two distinct statistical techniques of current interest for genetic data.

We first consider finite Bayesian mixture models. Mixture models are thought to have been first used by **?**, and are now widely used in genetics. They can be used to model sub-populations within a data set, which may arise when collecting human participants for a genetic study (for example, **?**). These sub-populations may be different ethnic groups, males and females, or even represent unknown heterogeneity. Another use of mixture models is to allow semi-parametric modelling in situations where the underlying distribution of the data may not be known. Mixture models are introduced in Chapter **??**, and in particular the *label switching* problem, occurring in Bayesian mixture models, is introduced. The label switching problem then provides the focus of **?**, which is presented in Chapter **??**. In **?**, existing methods to deal with the label switching problem are considered, and new probabilistic algorithms are introduced; the new and existing methods are then compared.

The second area considered is the recovery of direct effects from amongst a large number of predictors. Potential predictors in genetics include the 3 million SNPs that are available (**?**). With so many predictors under consideration, it is desirable to collect large samples to fully understand the statistical connections. High cost and low availability of participants, however, prohibits this. This leads to situations in which the number of predictors, p, is much larger than the number of observations, n. This so-called '$p \gg n$' problem causes traditional statistical methods, such as standard linear regression, to fail. Therefore, many new methods have been developed to deal with this problem, which are reviewed in Chapter **??**.

An issue closely related to the '$p \gg n$' problem is that neighbouring SNPs on the genome are often highly correlated (in linkage disequilibrium) with each other; this means that if the state of one SNP is known, the states of neighbouring SNPs can be predicted with high confidence. This correlation between neighbours causes a problem called *multicollinearity*. The multicollinearity issue, besides potentially causing similar kinds of degeneracies as the '$p \gg n$' problem, can lead to consistency being impossible to achieve. For example, consider two SNPs that are perfectly correlated, one of which possessing a true effect on some disease; then it is impossible to determine which of the two SNPs possesses the effect. This is less of an issue if the goal is to produce a good predictive model — in which case it is irrelevant whether the correct SNPs are included, provided accurate predictions can be made. The interest here, however, is in identifying the correct SNPs, which means that the uncertainty caused by issues such as multicollinearity must be dealt with. The appropriate way to deal with such uncertainty, we believe, is to provide a list of potential SNPs for the true origin of each effect, along with a list of probabilities that each of these SNPs is the true origin.

The '$p \gg n$' issue and the multicollinearity issue have inspired the development of the new method of direct effect testing (DET). DET is introduced and described in **?**, presented here in Chapter **??**, and its performance is illustrated and compared with existing methods. A follow-up comparison paper is given in Chapter **??** (**?**), where we analyse the DET method further. Further extensions of the DET method that have not yet been written for publication are presented in Chapter **??**.

Both the probabilistic relabelling algorithms and the DET approach have applications outside of genetics, and we discuss these in Chapter **??**, which also includes discussion of the future directions of this work.

Chapter 2

Bayesian Mixture Models

2.1 Introduction and Notation

Finite mixture modelling is a tool to model population heterogeneity, and a form of semi-parametric modelling. The applicability of mixture models has increased in the last 30 years, due to advances in methods for handling missing data and the exponential increase in available computing power. The missing data problem in mixture models can be solved using the EM algorithm (**?**); this allows robust, reliable maximum likelihood estimation in the presence of missing data. The EM algorithm can also be used in a Bayesian framework, by penalising the likelihood according to the desired prior (**?**). In the Bayesian setting, however, the real breakthrough came with the application of Markov chain Monte Carlo (MCMC) methods to mixture models (see, for example, **?**). MCMC allows for exploration of posterior and predictive surfaces of mixture models, both in great detail and with high efficiency.

Mixture models consider the population of observations as consisting of a number of sub-populations. Observations within a sub-population are assumed to be homogeneous, whilst observations in different sub-populations are assumed to be heterogeneous. To illustrate this idea, we introduce an example data set. The galaxy data was collected by **?**, and is now commonly used as an illustration of mixture models, and a test data set for new techniques. Figure **??** includes a histogram displaying the velocities at which 82 galaxies are travelling away from our own. There were 83 galaxies in the original paper, but one of the galaxies is commonly omitted from the data set (see **?**, and the references therein). The motivation for fitting a mixture model to this data set is that the galaxies are thought to be arranged into clusters, and within each cluster the velocities of the galaxies follow a given distribution. A normal distribution is often assumed, although student-t distributions, for example, have also been considered (**?**). In Figure **??** a density estimate arising from fitting a normal mixture distribution (assuming that there are four clusters) to the galaxy data is overlaid.

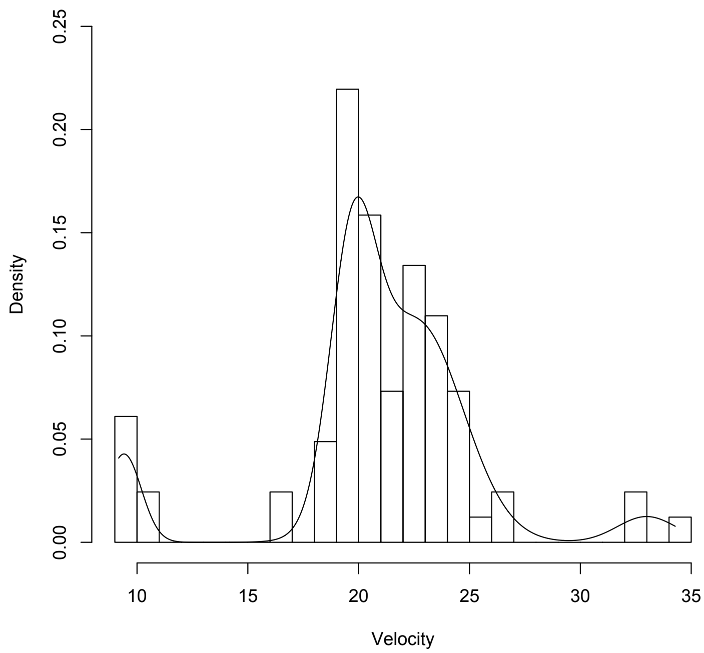

Figure 2.1: Histogram of the velocities of 82 galaxies, with normal mixture density added

In this Chapter we will begin by introducing MCMC, for the reader less familiar with the technique (Section **??**). We then introduce the method of Bayesian estimation of a mixture model by MCMC techniques (Section **??**), before briefly introducing the label switching problem (Section **??**), which is the subject of the paper following in Chapter **??**. First, the notation for this part of the thesis is introduced.

Suppose $Y_1, \ldots, Y_n$ is a random univariate sample of size n, taken from a population with probability density function (pdf)

$$p(Y) = \pi_1 f_1(Y) + \pi_2 f_2(Y) + \ldots + \pi_K f_K(Y), \tag{2.1}$$

with $K \geq 1$ (although $K = 1$ is a degenerate case), $\pi_k > 0$ ($k = 1, 2, \ldots, K$), $\sum_{k=1}^{K} \pi_k = 1$, and each $f_k(\cdot)$ is a pdf itself ($k = 1, 2, \ldots, K$). Then the population has a *finite mixture distribution*, and $p(\cdot)$ is a *finite mixture density function*; $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$ is the vector of *mixing weights* and the f_k's are called the *component densities.* It is commonly assumed that the f_k's all have the same distributional form, with mixture-specific parameters $\boldsymbol{\theta}_k$ and global parameters $\boldsymbol{\eta}$. Under these assumptions, Equation (**??**) becomes

$$p(Y|\boldsymbol{\gamma}) = \sum_{k=1}^{K} \pi_k f_k(Y|\boldsymbol{\theta}_k, \boldsymbol{\eta}),$$

where $\boldsymbol{\gamma} = (\boldsymbol{\pi}'; \boldsymbol{\theta}_1', ..., \boldsymbol{\theta}_K'; \boldsymbol{\eta}')'$, the vector of all defined parameters.

A convenient parameterisation is to introduce latent variables, Z_i, for $i = 1, \ldots, n$, where '$Z_i = k$' means that observation Y_i comes from sub-population k. Each Z_i is therefore distributed *a-priori*

$$Z_i \sim \text{Multinomial}\big(\mathbf{1}, (\pi_1, ..., \pi_K)'\big),$$

where $\mathbf{1}$ denotes the unitary vector of length K. The distribution of each Y_i, conditional on the value of the corresponding latent variable Z_i, is then the density of the corresponding component,

$$\mathrm{Y}_i|(Z_i = k) \sim f_k(\cdot|\boldsymbol{\theta}_k).$$

This interpretation can be verified by integrating out the Z_i's. When mixture distributions are used semi-parametrically, the $\boldsymbol{Z}$-variables do not have the same real-world interpretation as in the sub-population modelling case, but are useful tools in calculation nonetheless.

Previously, mixture model inference was carried out with a fixed number of components K (see, for example, **?**). If K was unknown, it was necessary to repeat the inference for different values of K then compare the resulting models to make a choice for K, using a model selection technique such as Akaike's information criterion (AIC) (**?**) or Schwarz's information criterion (BIC) (**?**). The introduction of variable dimension samplers for mixture models (**??**) mean it is now possible to treat K directly as an additional unknown parameter.

?, **?** and **?** are all good introductions to mixture models and their applications.

2.2 Bayesian Inference by MCMC

It is assumed that the reader is familiar with the Bayesian paradigm (otherwise see **?**, for an introduction). Recall that Bayesian inference involves postulating a prior distribution for each parameter in the model, representing available knowledge (or ignorance) before the data is collected. A posterior distribution $q(\boldsymbol{\gamma}|Y_1, \ldots, Y_n)$ is then calculated by combining the prior information with the information obtained from the data (i.e. the likelihood). Often, we then wish to perform inference on the posterior expectation of a function $h(\boldsymbol{\gamma})$, which involves calculating

$$E\big[h(\boldsymbol{\gamma})\big] = \int h(\boldsymbol{\gamma}) q(\boldsymbol{\gamma}|Y_1, \ldots, Y_n) d\boldsymbol{\gamma}. \tag{2.2}$$

It is not always possible to evaluate such an integral analytically. To solve this problem, one can instead draw a sequence of realisations $\boldsymbol{\gamma}^{(1)}, \ldots, \boldsymbol{\gamma}^{(R)}$ from the posterior distribution $q(\boldsymbol{\gamma}|Y_1, \ldots, Y_n)$, for some large number R, then approximate Equation (**??**) by taking the ergodic average

$$E[h(\boldsymbol{\gamma})] \approx \frac{1}{R} \sum_{r=1}^{R} h(\boldsymbol{\gamma}^{(r)}).$$

The *ergodic theorem* ensures that this approximation converges almost surely to the required expectation as $R \to \infty$. A popular method of generating the sequence $\boldsymbol{\gamma}^{(1)}, \ldots, \boldsymbol{\gamma}^{(R)}$ is Markov chain Monte Carlo (MCMC). The idea of MCMC is to construct an ergodic Markov chain that has the posterior distribution as its stationary distribution.

One algorithm to construct a Markov chain with the appropriate stationary distribution is the Gibbs sampler (see, for example, **?**), which works according to the following procedure. For simplicity, suppose that there are p parameters, denoted by $\gamma_1, \ldots, \gamma_p$. Let $\boldsymbol{\gamma}_{-j}$ denote the vector of all parameters except γ_j.

1. Initialise all parameters (typically by drawing their values from their priors, but other

initialisations are possible).

2. For $j = 1, \ldots, p$, update the parameter γ_j by drawing a new value from the full conditional distribution $q(\gamma_j|\boldsymbol{\gamma}_{-j}; Y_1, \ldots, Y_n)$. Often, the one-dimensional full conditional distributions are tractable, despite the full posterior distribution being intractable.

3. Record all new values of the parameters.

4. Iteratively repeat steps 2 and 3, a large number of times R.

In an attempt to ensure that only realisations of the parameter vector $\boldsymbol{\gamma}$ that are drawn from the posterior distribution are retained, the first few realisations, believed to have been drawn before the Markov chain was in equilibrium, are discarded. The discarded realisations are known as burn-in. Besides this, using Markov chains results in serially correlated observations (since every realisation depends on the previous realisation). To alleviate this problem, the recorded values for $\boldsymbol{\gamma}$ can be thinned, that is, only every few values are retained. See **?** for a more in-depth discussion of the issues surrounding the use of MCMC.

In this work, we have run the Gibbs sampler for 60000 iterations, and discarded a burn-in of 10000 realisations. We use a thinning factor of 10, i.e. only the values from every tenth iteration are retained.

2.3 MCMC and Bayesian Mixture Models

In this thesis we focus on mixture model inference in a Bayesian setting, using Gibbs sampling. A common method for conducting such inference is by using the latent variable structure given in Section **??**. This was first introduced by **?**.

For purposes of illustration, we will now give a specific example. It is assumed that the component densities (f_k's) are normal, and the 'random beta model' of **?** is used to construct a hierarchical Bayesian model, which is specified below. The random beta model assumes that none of the parameters are known in advance. For a given number of components K, the distribution of the mixture model is given by

$$Y_i|(\boldsymbol{\gamma}, K) \sim \sum_{k=1}^{K} \pi_k N(\mu_k, \sigma_k^2),$$

so the observations follow a mixture of normal distributions. Conditional on its latent variable, Z_i, each Y_i ($i = 1, \ldots, n$) is then distributed according to the corresponding normal distribution from the mixture distribution,

$$Y_i|(\boldsymbol{\gamma}, K, Z_i = k) \sim N(\mu_k, \sigma_k^2).$$

As this is a Bayesian model, prior distributions are needed for all the parameters. Indeed, a *hyper-parameter*, ϕ, is introduced that is used in the priors for the parameters of the model, but has a prior distribution itself. The priors for the random beta model are given by

$$\begin{aligned}
\mu_k &\sim N(\xi, \kappa^{-1}), \\
\sigma_k^{-2}|\phi &\sim \text{Gamma}(\alpha, \phi), \\
\phi &\sim \text{Gamma}(g, h), \\
\boldsymbol{\pi} &\sim \text{Dirichlet}(\boldsymbol{\delta}).
\end{aligned}$$

To provide values for the constants in the priors, if R is the range of the data, we choose $\alpha = 2$, $g = 0.2$, $h = 10/R^2$, $\kappa = 4/R^2$, $\boldsymbol{\delta} = \mathbf{1}$, the unitary vector of length K, and ξ as the midpoint of the minimum and maximum values of the data. These choices are in line with the values used by **?**, see their paper for further discussion. These priors are hence designed to be non-informative. It is impossible to place improper priors on any of the parameters, since in the case of mixture models this will generate an improper posterior. The improper posterior can arise because there is a positive probability of a component being empty (i.e. containing no observations), in which case it is solely the prior that generates the posterior for any parameters specific to that component (**?**).

The full conditional distributions for the Gibbs sampler are then as follows (where '$|\boldsymbol{\gamma}_-$'

is used to denote conditioning on all remaining variables in $\boldsymbol{\gamma}$):

$$
\begin{aligned}
Z_i|\boldsymbol{\gamma}_- &\sim \text{Multinomial}\big(\mathbf{1}, (\tau_{i1}, ..., \tau_{iK})'\big), \\
\phi|\boldsymbol{\gamma}_- &\sim \text{Gamma}\left(K\alpha + g, \sum \sigma_k^{-2} + h\right), \\
\sigma_k^{-2}|\boldsymbol{\gamma}_- &\sim \text{Gamma}\left(\alpha + \frac{n_k}{2}, \phi + \frac{\sum_{i:Z_i=k}(X_i - \mu_k)^2}{2}\right), \\
\mu_k|\boldsymbol{\gamma}_- &\sim N\left(\frac{\kappa\xi + \sigma_k^{-2}\sum_{i:Z_i=k} X_i}{\sigma_k^{-2} n_k + \kappa}, (\sigma_k^{-2} n_k + \kappa)^{-1}\right), \\
\boldsymbol{\pi}|\boldsymbol{\gamma}_- &\sim \text{Dirichlet}\big((\delta_1 + n_1, ..., \delta_K + n_K)'\big),
\end{aligned}
$$

where the Z_i's are the latent allocation variables, $\tau_{ik} \propto \pi_k \times$ likelihood$(Y_i|Z_i = k)$, and n_k is the number of observations allocated to the k^{th} component.

The number of components K is also treated as an object of inference in most situations; the choice of prior for K, however, is a delicate issue (**?**). Since the priors for the model components do not depend on K, updating K does not affect the updates to the other parameters within each fixed value of K. We will use a Poisson(1) prior on the number of components K, which is argued for in **?**. The number of components K will be updated using the birth-death sampler suggested by **?**, which we now briefly outline.

The components of the mixture model are viewed as point processes on the space $[0, 1] \times \boldsymbol{\Theta}$, where each mixing weight π_k is in the interval $[0, 1]$ and the mixture-specific components $\boldsymbol{\theta}_k$ belong to the parameter space $\boldsymbol{\Theta}$. Denoting the current situation of the point process by x, where there are currently K components,

$$
x = \big\{(\pi_1, \boldsymbol{\theta}_1), \ldots, (\pi_K, \boldsymbol{\theta}_K)\big\}.
$$

A *birth* adds a new component to the mixture model, causing the point process to move to a new location

$$
x_b = x \cup (\pi_{K+1}, \boldsymbol{\theta}_{K+1}) = \left\{\big(\pi_1(1 - \pi_{K+1}), \boldsymbol{\theta}_1\big), \ldots, \big(\pi_K(1 - \pi_{K+1}), \boldsymbol{\theta}_K\big), \big(\pi_{K+1}, \boldsymbol{\theta}_{K+1}\big)\right\},
$$

where the mixing weight of the new component, π_{K+1}, is drawn from a Beta$(1, K)$ distribution and the mixture-specific parameters for the new component are drawn from their prior distributions. Births occur at a fixed rate λ_b.

The *death* of the k^{th} component causes the point process to move to a new location

$$x_d = x \setminus (\pi_k, \boldsymbol{\theta}_k) = \left\{ \left(\frac{\pi_1}{1-\pi_k}, \boldsymbol{\theta}_1 \right), \ldots, \left(\frac{\pi_K}{1-\pi_k}, \boldsymbol{\theta}_K \right) \right\}.$$

The death of the k^{th} component occurs at a variable rate $\delta_k(x)$, given by

$$\delta_k(x) = \lambda_b \frac{L\left\{x \setminus (\pi_k, \boldsymbol{\theta}_k)\right\}}{L(x)} \frac{q(K-1)}{Kq(K)},$$

where L denotes the likelihood and $q(\cdot)$ is the prior on K. 'Useful' components are unlikely to die, whereas 'useless' components will be killed very quickly. The death rates depend on the current state of the point process, x, so are updated after each new birth or death.

This birth-death process is run for a fixed time on each iteration of the MCMC, to update the number of components K. Without loss of generality, the length of the birth-death process can be fixed to one time unit. In this work the birth-death process was initialised at the beginning of each MCMC sampler with one component, $K = 1$. The birth rate was chosen to be $\lambda_b = 2$. For further explanations, and proofs of the validity of the birth-death procedure, see **?**. Alternative algorithms for updating K include the reversible jump MCMC procedure described in **?**.

The MCMC procedure for mixture model inference is then as follows on each iteration:

1. Run the birth-death process to update the number of components K.

2. Update the Z_i's by sampling from their full conditional distributions, which can be thought of as allocating the observations to components.

3. Update the hyper-parameter ϕ.

4. Update the component-specific parameter vectors $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$ and $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_K^2)$.

2.4 The Label Switching Problem

Label switching refers to the fact that the order that the component densities are in can change during an MCMC run. For example, if there are two components ($K = 2$), the chain may begin with $k = 1$ corresponding to what we intuitively think of as the first component, and $k = 2$ the second component. Then at some point during the MCMC the roles could be reversed, so that $k = 1$ now corresponds to our notion of the second component, and $k = 2$ the first component. Label switching becomes a problem when one is interested in obtaining component-specific information from an MCMC sample on a mixture distribution. Examples of component-specific information we may require include the posterior distributions of the parameters of each component. Formally, label switching occurs because for any possible permutation $\nu(\cdot)$, of the component labels $\{1, 2, \ldots, K\}$, the likelihood of the mixture distribution satisfies

$$p(Y|\boldsymbol{\gamma}) = \sum_{k=1}^{K} \pi_k f(Y|\boldsymbol{\theta}_k, \boldsymbol{\eta}) = \sum_{k=1}^{K} \pi_{\nu(k)} f(Y|\boldsymbol{\theta}_{\nu(k)}, \boldsymbol{\eta}). \tag{2.3}$$

If the prior distributions are also exchangeable (i.e. containing no component-specific information), the posterior distribution inherits the same property, and hence possesses $K!$ symmetric modes. This is a problem as an MCMC sampler may move from one of these modes to another between iterations, resulting in a label switch.

The label switching problem is described more formally in Chapter **??**, whereas here a simple illustration of the problem is presented. Suppose 200 realisations are generated from the two component mixture distribution

$$0.5\big\{N(0, 1) + N(4, 1)\big\},$$

so that $\mu_1 = 0$ is the true value of the mean of the first component, and $\mu_2 = 4$ is the true value of the mean of the second component.

Figure **??** gives 1000 iterations of the output of the MCMC sampler for the component means μ_1 and μ_2. The top panel of the Figure demonstrates what happens when we ignore the possibility of label switching. Many times during the sampler, the roles of the first and second component have switched. This makes any component-specific inference, such as calculating parameter estimates, impossible. The lower panel of the figure shows the paths of the component means after an algorithm to 'relabel' the output from the MCMC has been applied, in an attempt to remove any label switches that have occurred. In fact, the relabelling algorithm applied here was a simple identifiability constraint (see Section **??**). In this example it is obvious at any given iteration which component is which, so the relabelling could easily have been carried out by eye.

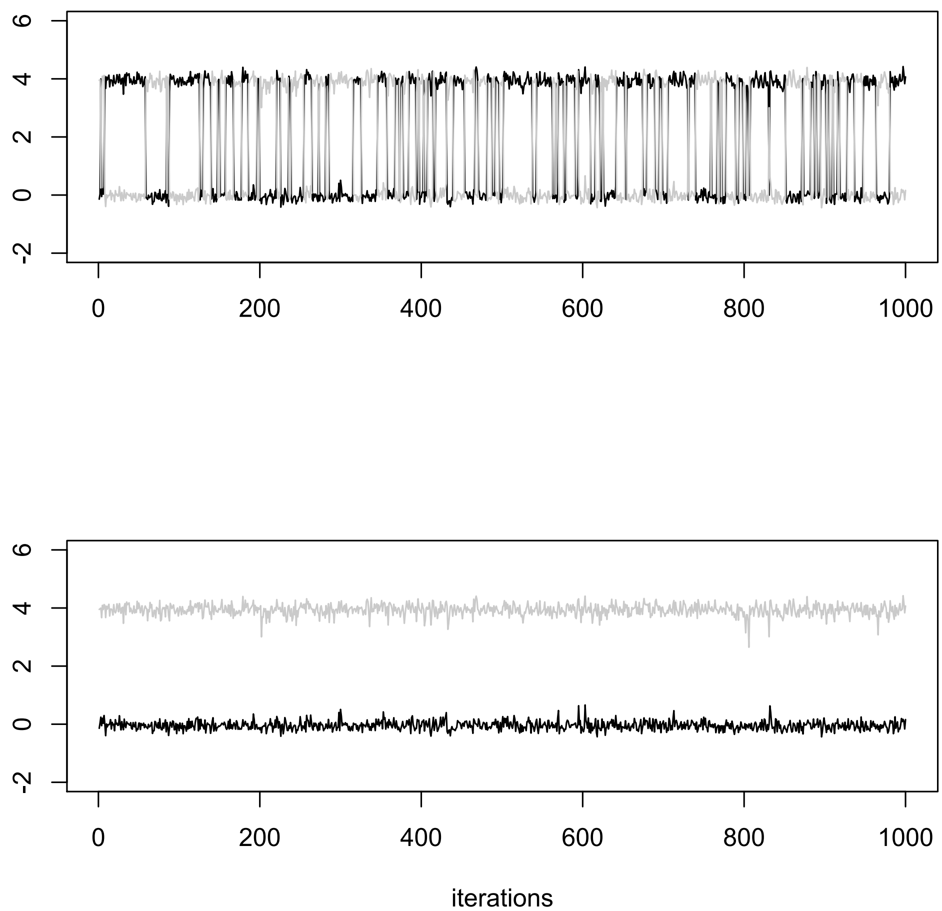

Figure 2.2: Graphs showing the paths of μ_1 (black line) and μ_2 (grey line) for 1000 iterations of an MCMC sampler. Top panel — no relabelling; bottom panel — after relabelling.

The label switching problem is far more pronounced in situations where the mixture components are harder to distinguish, because we cannot identify if or when a label switch has occurred. Hence, advanced algorithms are needed to deal with the label switching issue in an automated yet sensible way. Such algorithms are called 'relabelling' algorithms. In Chapter **??**, which is the paper **?**, we review existing relabelling algorithms, introduce new probabilistic algorithms and compare the performance of all the algorithms.

Chapter 3

Probabilistic Relabelling Strategies for the Label Switching Problem in Bayesian Mixture Models

Abstract

The label switching problem is caused by the likelihood of a Bayesian mixture model being invariant to permutations of the labels. The permutation can change multiple times between Markov chain Monte Carlo (MCMC) iterations making it difficult to infer component-specific parameters of the model. Various so-called 'relabelling' strategies exist with the goal to 'undo' the label switches that have occurred to enable estimation of functions that depend on component-specific parameters. Most existing approaches rely upon specifying a loss function, and relabelling by minimising its posterior expected loss. In this paper we develop probabilistic approaches to relabelling that allow estimation and incorporation of the uncertainty in the relabelling process. Variants of the probabilistic relabelling algorithm are introduced and compared to existing loss function based methods. We demonstrate that the idea of probabilistic relabelling can be expressed in a rigorous framework based on the EM algorithm.

Keywords: Bayesian, Identifiability, Label switching, MCMC, Mixture model.

3.1 Introduction

Mixture models have been used as tools to model heterogeneity for over 100 years. Developments in Markov chain Monte Carlo (MCMC) methods (see, for example, **?**) opened the door for mixture models in a Bayesian framework as they allow efficient exploration of posterior and predictive surfaces of these models. The use of these Bayesian mixture models has given rise to new problems, particularly when estimating component-specific parameters of the model and interpreting marginal posterior densities.

The label switching problem arises as the components of the Bayesian mixture model can be ordered arbitrarily. During one run of an MCMC sampler, the order of components can change multiple times between iterations. To obtain a meaningful interpretation of the components it is necessary to account for these changes, which has been called *relabelling* (for example, **?**). Various functions of interest, such as recovery of the full mixture posterior and its associated moments, may be invariant to the labelling permutations. For this type of inference, the label switching problem need not concern us. On many occasions, however, it is of interest to infer parameters that are specific to individual components of the mixture model. This may be because the components of the model have some interpretation, in the sense of a one-to-one correspondence to true components in the population, or alternatively we may be using mixture models to carry out semi-parametric density estimation, and the purpose of the relabelling is to provide coherent estimates of the components that make up the density estimate. In either case, we must find methods to 'relabel' the results of an MCMC run so that the components are in the same order at each iteration.

A wide array of strategies exist in the literature for 'relabelling' MCMC output in an attempt to remove the label switching problem — we divide them here into three categories. *Identifiability constraints* involve relabelling the output of the MCMC sampler so that the posterior obtained satisfies a constraint on the component parameters. The constraint is chosen such that exactly one relabelling satisfies the constraint at each iteration of the sampler. *Deterministic relabelling algorithms* select a relabelling at each iteration of the MCMC output that minimizes the posterior expectation of some loss function. Naturally, a variety of loss functions have been considered by different authors. *Probabilistic* approaches are a relatively new idea in which one acknowledges that there is uncertainty in the relabelling that should be selected on each iteration of the MCMC output. In contrast, both identifiability constraints and deterministic relabelling algorithms assume that the relabelling that has been carried out is 'correct'.

The contribution of this paper is to develop and extend the idea of probabilistic relabelling, which was introduced originally in **?**. We frame probabilistic relabelling as an application of the EM algorithm, where the missing data is the order that the components are in at each iteration of the MCMC. Two novel probabilistic algorithms based on the stochastic EM (SEM) are developed.

We will proceed, in Section **??**, by briefly describing some of the relabelling algorithms currently available, before we introduce new strategies for probabilistic relabelling. Section **??** evaluates the performance of the strategies on observed as well as simulated data. We conclude with a discussion of the advantages and disadvantages of the various methods and some future directions in Section **??**.

3.2 Relabelling Strategies

Suppose n observations $Y_1, \ldots, Y_n$ are taken from a K-component mixture distribution where all the components have the same distributional form, with mixture-specific parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}'_1, \ldots, \boldsymbol{\theta}'_K)'$, global parameters $\boldsymbol{\eta}$ and mixing weights $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)'$, summarized by $\boldsymbol{\gamma} = (\boldsymbol{\pi}'; \boldsymbol{\theta}'_1, \ldots, \boldsymbol{\theta}'_K; \boldsymbol{\eta}')'$. The mixture distribution for a single observation Y_i is then given by

$$p\left(Y_i|\gamma\right) = \sum_{k=1}^{K} \pi_k f_k\left(Y_i|\boldsymbol{\theta}_k, \boldsymbol{\eta}\right),$$

with $K \geq 1$, $\pi_k > 0$ $(k = 1, 2, \ldots, K)$, $\sum_{k=1}^{K} \pi_k = 1$ and $f_k(\cdot|\boldsymbol{\theta}_k, \boldsymbol{\eta})$ is a density function parametrized by $\boldsymbol{\theta}_k$ and $\boldsymbol{\eta}$. For convenience we introduce latent variables $\boldsymbol{Z} = (Z_1, \ldots, Z_n)'$, where '$Z_i = k$' indicates membership of the observation Y_i to class k, with for $i = 1, \ldots, n$,

$$Z_i \overset{\text{i.i.d.}}{\sim} \text{Multinomial}\big(\mathbf{1}, \boldsymbol{\pi}\big),$$

where $\mathbf{1}$ denotes the unitary vector of length K. Conditional on belonging to class k, observation Y_i will be distributed according to $f_k(\cdot|\boldsymbol{\theta}_k, \boldsymbol{\eta})$,

$$Y_i|(Z_i = k) \sim f_k(\cdot|\boldsymbol{\theta}_k, \boldsymbol{\eta}).$$

Each Z_i is then an unknown categorical variable that denotes the sub-population from which observation Y_i originates. Bayesian inference can be conducted on such a model using MCMC (**?**). This proceeds, on each iteration r, by drawing a vector of component memberships $\boldsymbol{Z}^{(r)}$, and parameter estimates $\boldsymbol{\gamma}^{(r)}$, from the posterior. Throughout this paper, for ease of illustration we will assume that each $f_k(\cdot)$ is a normal distribution with mean μ_k and variance σ_k^2. For the priors we will use the hierarchical 'random beta' model in **?**, following their suggestions on the hyper-parameter choices. For the number of components K we use a Poisson(1) prior as argued for in **?**.

Let S_K denote the set of all permutations on $\{1, 2, \ldots, K\}$. The label switching problem arises because the likelihood

$$p\left(Y_1, \ldots, Y_n|\boldsymbol{\gamma}\right) = \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} \pi_{\nu(k)} f_{\nu(k)} \left(Y_i|\boldsymbol{\theta}_{\nu(k)}, \boldsymbol{\eta}\right) \right\}$$

is identical for all $\nu \in S_K$. If exchangeable priors are used (containing no component-specific information) then the posterior has the same property, resulting in the posterior surface having $K!$ symmetric modes, each associated with a different labelling permutation $\nu \in S_K$. This is problematic because each iteration of the MCMC sampler r, $r = 1, \ldots, R$, has an associated permutation $\nu^{(r)} \in S_K$. Then for $r_1 \neq r_2$, it may be that $\nu^{(r_1)} \neq \nu^{(r_2)}$, i.e. the sampler can move from one mode to another between iterations. This makes an ergodic average estimate of a component-specific parameter, for example,

$$E[\theta_1] \approx \frac{1}{R}\sum_{r=1}^{R} \theta_1^{(r)}, \tag{3.1}$$

somewhat meaningless. Indeed, if the chain is in equilibrium, then the estimate of $E[\theta_k]$ should be the same for all k, since such a chain explores equally all the symmetric modes.

The idea of *relabelling* the MCMC output is to account for the permutations $\nu^{(r)}$, $r = 1, \ldots, R$, in such a way that an ergodic average estimate such as Equation (**??**) is made meaningful. Of course, we generally have limited data, and can never say with certainty whether we truly have agreement $\nu^{(r_1)} = \nu^{(r_2)}$, for $r_1 \neq r_2$. Indeed, in our view the $\nu^{(r)}$s are themselves parameters with associated uncertainty. Define a *relabelled posterior*, $q^*(\cdot)$, as the posterior density that we obtain when we attempt to account for the permutations $\nu^{(r)}, r = 1, \ldots, R$, across the iterations of an MCMC sampler. This is not unique — firstly there are $K!$ versions of it that correspond to applying a permutation ν to the entire output of an MCMC to yield an equivalent answer. Secondly, we accept that it is not possible to find the 'correct' relabelled posterior due to the uncertainty in estimating the $\nu^{(r)}$s — we approximate this by the various relabelling methods considered in this paper. A version of the relabelled posterior is then useful when one conducts component-specific inference.

3.2.1 Identifiability Constraints

The first efforts to deal with the label-switching problem involve placing an *Identifiability Constraint* (IC) on the parameter space (see, for example, **?**). The idea is to define a restricted parameter space $\mathcal{A}$ such that there exists a unique permutation $\nu^* \in S_K$ that satisfies $\left(\boldsymbol{\theta}'_{\nu^*(1)}, \ldots, \boldsymbol{\theta}'_{\nu^*(K)}\right)' \in \mathcal{A}$, for component-specific parameters $\boldsymbol{\theta}_k, k = 1, \ldots, K$. The simplest example in the normal distribution case is the constraint $\mu_1 < \mu_2 < \ldots < \mu_K$, or the same constraint on the mixture proportions or component variances. More sophisticated alternatives can be found, for example, in **?**.

This approach is simple and works well in many situations. Proposition 3.1 of **?** demonstrates that the relabelling for such a strategy can be carried out after the MCMC has run, provided the priors are exchangeable. **?** notes that use of the IC leads, asymptotically in n, to the correct marginals for the true parameter vector $\boldsymbol{\theta}$ being recovered, provided $\boldsymbol{\theta} \in \mathcal{A}$. Nevertheless, for finite n it is found that the parameter estimates are 'pushed apart'; that is, the difference between the parameters of adjacent components is typically over-estimated (**?**). This is a consequence of the fact that we are effectively imposing *a-priori* that the joint prior of $\boldsymbol{\theta}$ must satisfy the constraint, despite originally imposing exchangeable priors, suggesting we know nothing to distinguish the components of the mixture model. Moreover, it can be difficult to find a sensible subspace, $\mathcal{A}$, when the mixture-specific parameters are multidimensional.

3.2.2 Deterministic Relabelling Algorithms

The idea of the relabelling algorithm is that we believe that the permutations $\nu^{(r_1)}$ and $\nu^{(r_2)}$ match (for $r_1 \neq r_2$; $r_1, r_2 \in \{1, 2, \ldots, R\}$) when a characteristic about the ${r_1}^{\text{th}}$ iteration under permutation $\nu^{(r_1)}$ is 'close' to that characteristic of the ${r_2}^{\text{th}}$ iteration under permutation $\nu^{(r_2)}$. There is a vast literature on the application of such algorithms to the label switching problem, all considering different characteristics about each iteration on which to measure closeness, and how one does measure closeness. **?** and **?** give methods where the characteristic is the estimates of the parameters on each iteration r, $\boldsymbol{\theta}^{(r)}$. **?** produces a method in which the characteristic is the matrix of allocation probabilities of the observations to each component of the mixture, $\boldsymbol{P}^{(r)}$ whilst **?** measure closeness in the allocation vector $\boldsymbol{Z}^{(r)}$.

Call the characteristic on which we measure closeness C, and the measure of closeness between two characteristics at iterations r_1 and r_2 as $L(C^{(r_1)}, C^{(r_2)})$, which is a loss function that is large when the discrepancy between $C^{(r_1)}$ and $C^{(r_2)}$ is large. When we apply a permutation $\nu^{(r)}$ to iteration r we will write $\nu^{(r)}(C^{(r)})$.

We are not interested *per se* in pairwise closeness, but closeness of the characteristics across the entire MCMC sample, $\{C^{(1)}, \ldots, C^{(R)}\}$, as we wish the entire sample to be relabelled 'correctly'. To take this into account in an efficient manner, many of the relabelling algorithms adopt a K-means style approach, which can be described in a general manner as follows:

1. Choose C to minimize $\sum_{r=1}^{R} L\left\{C, \nu^{(r)}(C^{(r)})\right\}$. In the K-means analogy, view C as the centroids of the clusters. In common with this analogy, C is usually calculated as the ergodic average of the characteristics $C^{(r)}$, $r = 1, \ldots, R$.

2. For $r = 1, \ldots, R$ choose $\nu^{(r)}$ to minimize $L\left\{C, \nu^{(r)}(C^{(r)})\right\}$, which is equivalent to allocating the observations to the clusters. **?** demonstrates that it is usually possible to achieve this quickly, using a variant of the transportation algorithm.

3. Repeat 1 and 2 until an optimal solution is reached.

The algorithm should be run from multiple starting positions (initial permutations of the MCMC iterations) as it is only guaranteed to converge to a local maximum rather than the global maximum (see, for example, **?**). The approach corresponds to minimising the approximate posterior expectation of the loss function L, with the approximation arising from averaging over the MCMC output. The iterative nature of the algorithm means that it must be run after the MCMC has completed.

ICs and relabelling algorithms have very similar goals, in that they assign meaning to each of the components. For example, under the IC considered above when we talk about the first component we mean 'the component with the smallest mean'. Relabelling algorithms attempt to give components meaning by enforcing some form of stable behaviour between iterations of the MCMC. If the goal of the inference is parameter estimation, it seems sensible to use an algorithm that stabilises the relabelled posterior of the parameters, using for example the algorithm of **?**. **?**, however, takes the opposing view that one should relabel using a different feature than the one of statistical interest, for example, relabel based on component allocations when interested in parameter estimates.

A separate class of algorithms are the label invariant loss function approaches introduced by **?**. Here, the idea is to measure closeness between iterations of the MCMC without relying on labelling information. For example, one could consider pairwise comparison of the allocation of observations to components (**?**).

3.2.3 Probabilistic Relabelling Algorithms

Probabilistic relabelling was first introduced by **?**. The idea is that the permutation $\nu^{(r)}$ that is associated with the r^{th} iteration of the MCMC sampler is unknown. Therefore, the permutation can be viewed as having the discrete density $g_r(\nu^{(r)}; \boldsymbol{\gamma}, \boldsymbol{Y})$ over $\nu^{(r)} \in S_K$, conditional on the data, $\boldsymbol{Y} = (Y_1, \ldots, Y_n)'$, and the full vector of parameters, $\boldsymbol{\gamma}$. **?** then shows, using the strong law of large numbers, that one can estimate a quantity of interest $h(\cdot)$ via

$$h(\boldsymbol{\gamma}) = \frac{1}{R} \sum_{r=1}^{R} \sum_{\nu^{(r)} \in S_K} h\left\{\nu^{(r)}(\boldsymbol{\gamma}^{(r)})\right\} \hat{g}_r(\nu^{(r)}; \hat{\boldsymbol{\gamma}}, \boldsymbol{Y}), \tag{3.2}$$

where $\nu^{(r)}(\boldsymbol{\gamma}^{(r)})$ represents the parameter vector with the component-specific parameters permuted by $\nu^{(r)}$. The function of interest $h(\cdot)$ may depend additionally or alternatively on the allocation vector $\boldsymbol{Z}$.

To use this approach we need a way to estimate $g_r(\cdot)$, and we also need to know in advance the vector of true parameters $\boldsymbol{\gamma}$. **?** gives various suggestions on how each of these issues may be dealt with. For example, the parameters $\boldsymbol{\gamma}$ can be derived by averaging over a small number of iterations from the MCMC, determined by eye not to have switched labels. The permutation densities $g_r(\cdot)$ are derived by estimating the posterior surface of the relabelled posterior using again a small number of iterations where the labels are deemed not to have switched. This uses a normal approximation, and the idea of estimating the relabelled posterior to deal with label switching was first suggested by **?**.

Next we introduce a novel approach to probabilistic relabelling, in which $g_r(\cdot)$ and $\boldsymbol{\gamma}$ are estimated in an iterative fashion. An EM-type approach is adopted, where the missing data are the permutations $\{\nu^{(r)}, r = 1, \ldots, R\}$ applied at each stage. The permutation densities, $g_r(\cdot)$, are estimated by conditioning only on the data, $\boldsymbol{Y}$, the current estimate of the parameters, $\boldsymbol{\gamma}$, and the current allocation vector, $\boldsymbol{Z}^{(r)}$. Letting $S_k^r = \{i : z_i^{(r)} = k\}$ be the set of indices of the observations belonging, before permutation, to the k^{th} parameter at iteration r, we calculate

$$\hat{g}_r(\nu^{(r)}; \hat{\boldsymbol{\gamma}}, \boldsymbol{Y}, \boldsymbol{Z}^{(r)}) \propto \prod_{k=1}^{K} \prod_{i \in S_k^r} \hat{\pi}_{\nu(k)} f_{\nu(k)} \left(Y_i | \hat{\boldsymbol{\theta}}_{\nu(k)}, \hat{\boldsymbol{\eta}}\right), \tag{3.3}$$

where the right hand side corresponds to the allocated likelihood. So rather than using a normal approximation to the surface of the relabelled posterior, $g_r(\cdot)$ is estimated based on the allocated likelihood of the data under each permutation, the current estimate of the parameters (permuted according to the permutation under consideration) and the current allocation vector $\boldsymbol{Z}^{(r)}$. Finally $\hat{g}_r(\cdot)$ is normalised to sum to one over all possible permutations. A detailed derivation of Equation (**??**) is given in the Appendix.

The usual application of the EM algorithm (**?**) to the mixture problem views the available data as the observations, and the missing data the membership of the observations to the various components. The framework introduced here, on the other hand, can be interpreted as an EM algorithm where the available data are the output from the MCMC sampler, and the missing data are the permutations $\{\nu^{(r)}, r = 1, \ldots, R\}$ applied at each stage. One could loosely consider the approaches suggested by **?** as corresponding to a single iteration of such an EM algorithm, with sensible starting values chosen. We propose now a variety of extensions and alternatives that stem from placing probabilistic relabelling in this framework. We suggest first an iterative EM algorithm, which proceeds, after initialising estimates of the parameters $\boldsymbol{\gamma}$ using, for example, an IC, by:

E Step Estimate the densities $\{g_r(\cdot), r = 1, \ldots, R\}$ using the current estimate of $\boldsymbol{\gamma}$, via Equation (**??**).

M step Update estimates of $\boldsymbol{\gamma}$ using Equation (**??**), with appropriate choices of $h(\cdot)$. For example, the estimate of the component weight π_1 may be updated by

$$\hat{\pi}_1 = \frac{1}{R} \sum_{r=1}^{R} \sum_{\nu^{(r)} \in S_K} \pi_1^{(r)} \hat{g}_r(\nu^{(r)}; \hat{\boldsymbol{\gamma}}, \boldsymbol{Y}, \boldsymbol{Z}^{(r)}).$$

As with all EM-type algorithms, convergence to the global maximum is not guaranteed — local modes or saddle points may instead be found. Therefore it is advised to use multiple starting points (different estimates of $\boldsymbol{\gamma}$). We call this EM approach 'EMP' (EM based probabilistic relabelling).

A popular alternative to the EM algorithm is the stochastic EM algorithm (SEM) (**?**). This introduces an extra step 'the S step', where the missing data is simulated from its estimated density. This constitutes drawing $\nu^{(r)}$ multinomially from the discrete density $g_r(\cdot)$. The randomness that this modification introduces helps to avoid the algorithm getting caught in local modes, and provides faster convergence. Additionally, the convergence of the SEM does not depend on the starting position (**?**). A SEM-type probabilistic relabelling strategy is as follows:

E step Estimate the densities $\{g_r(\cdot), r = 1, \ldots, R\}$ using the current estimate of $\boldsymbol{\gamma}$, via Equation (**??**).

S step Simulate values for the permutations $\{\nu^{(r)}, r = 1, \ldots, R\}$ by drawing multinomially from the corresponding densities $g_r(\cdot)$, calling these simulated permutations $\tilde{\nu}^{(r)}$, $r = 1, \ldots, R$.

M step Update estimates of $\boldsymbol{\gamma}$ by taking ergodic averages over the sample after accounting for the permutations $\tilde{\nu}^{(r)}$, $r = 1, \ldots, R$. For example, the estimate of the component weight π_1 may be updated by
$$\hat{\pi}_1 = \frac{1}{R}\sum_{r=1}^{R} \pi_1^{(r)},$$
after the inverse of $\tilde{\nu}^{(r)}$ has been applied at each r.

We call this approach 'SEMP' (SEM based probabilistic relabelling).

A final alternative that we suggest acknowledges that $\boldsymbol{\gamma}$ is itself unknown. We consider estimating the permutation densities $g_r(\cdot), r = 1, \ldots, R$, without conditioning on $\boldsymbol{\gamma}$ by integrating $\boldsymbol{\gamma}$ out with respect to its *relabelled* posterior, $q^*(\boldsymbol{\gamma})$:

$$\hat{g}_r(\nu^{(r)}; \boldsymbol{Y}, \boldsymbol{Z}^{(r)}) \propto \int \prod_{k=1}^{K} \prod_{i \in S_k^r} \hat{\pi}_{\nu(k)} f_{\nu(k)} \left(Y_i | \hat{\boldsymbol{\theta}}_{\nu(k)}, \hat{\boldsymbol{\eta}} \right) q^*(\boldsymbol{\gamma}) d\hat{\boldsymbol{\gamma}},$$

and approximate the integral by the Monte Carlo estimate over the MCMC sample

$$\hat{g}_r(\nu^{(r)}; \boldsymbol{Y}, \boldsymbol{Z}^{(r)}) \propto \frac{1}{R}\sum_{r=1}^{R} \left\{ \prod_{k=1}^{K} \prod_{i \in S_k^r} \pi_{\nu(k)}^{(r)} f_{\nu(k)} \left(Y_i | \boldsymbol{\theta}_{\nu(k)}^{(r)}, \boldsymbol{\eta}^{(r)} \right) \right\}. \tag{3.4}$$

This leads to the algorithm:

E step Estimate the densities $\{g_r(\cdot), r = 1, \ldots, R\}$ using the current estimate of the relabelled posterior density of $\boldsymbol{\gamma}$, via Equation (**??**).

S step Simulate values for the permutations $\{\nu^{(r)}, r = 1, \ldots, R\}$ by drawing multinomially from the corresponding densities $g_r(\cdot)$, calling these simulated permutations $\tilde{\nu}^{(r)}$, $r = 1, \ldots, R$.

M step Estimate the relabelled posterior density, $q^*(\boldsymbol{\gamma})$, using the output from the MCMC and the current estimates $\tilde{\nu}^{(r)}$, $r = 1, \ldots, R$.

The M step is, therefore, fundamentally different from a usual EM or SEM algorithm — we estimate an entire posterior rather than point estimates of the parameters. We call this approach 'SEMUP' (SEM based unconditional probabilistic relabelling).

3.2.4 Comments

For the remainder of the paper we will consider seven different relabelling strategies, the IC, three deterministic relabelling algorithms and the three variants of probabilistic relabelling we introduced in the previous section. The notation used for the methods is defined in Table **??**.

Table 3.1: Relabelling algorithms evaluated

Notation	Method	Source
IC	Identifiability constraint	**?**
PL	Parameter relabelling algorithm	**?**
CPL	Class probability relabelling algorithm	**?**
AL	Allocation vector relabelling algorithm	**?**
EMP	EM probabilistic	Section **??**
SEMP	SEM probabilistic	Section **??**
SEMUP	SEM unconditional probabilistic	Section **??**

One of the disadvantages of relabelling algorithms and ICs is that they apply a specific permutation $\nu^{(r)}$ at each iteration r, with no indication on how uncertain we are that this particular permutation is 'correct'. Using probabilistic methods, uncertainty in relabelling can be quantified by how close to one the probability of the most likely permutation being correct, $\max_{\nu^{(r)}} \left\{\hat{g}_r(\nu^{(r)}; \hat{\boldsymbol{\gamma}}, \boldsymbol{Y}, \boldsymbol{Z}^{(r)})\right\}$, is, for each iteration of the MCMC.

A further advantage of probabilistic relabelling is the improved recovery of the posterior tails, which are often truncated using other methods. Consider 50 simulated observations from $0.5N(0,1)+0.5N(2,1)$. Figure **??** compares the marginal posteriors for μ_1 (defined as the component mean with smallest ergodic average) under the PL and SEMP methods, assuming that all parameters in the model are unknown. The distributions are quite different in shape with the right hand tail being truncated for the PL algorithm in comparison to the SEMP method, which is compensated by a higher peak. Similar results are observed in all the probabilistic methods. This clearly shows the superior ability of probabilistic relabelling to recover posterior tails.

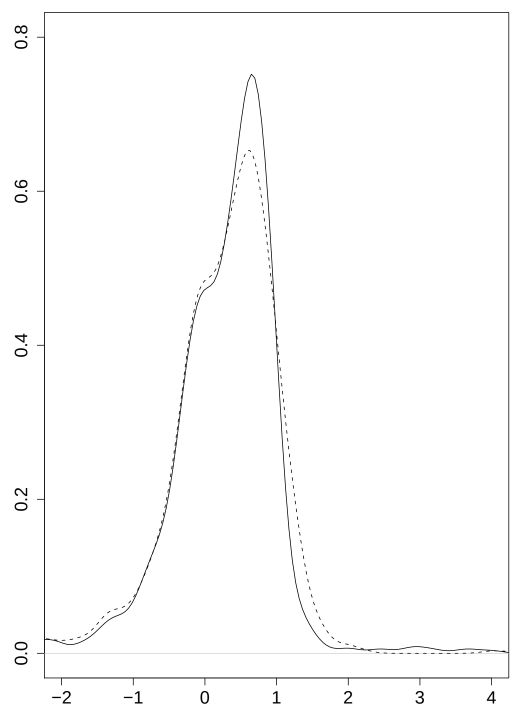

Figure 3.1: Graphs showing the posteriors for μ_1 (defined as the μ with smallest ergodic average) of the two component mixture model $0.5N(0,1) + 0.5N(2,1)$, for the PL (solid line) and SEMP (dashed line) algorithms

3.3 Comparison of Methods

To evaluate the proposed algorithms we will now compare them to existing methods on observed and simulated data. The seven relabelling strategies that will be compared are given in Table **??**.

3.3.1 The Galaxy Data

For the initial comparison we investigate the galaxy data which consist of the velocities of 82 different galaxies (**?**). A histogram of the data is given in Figure **??**. This dataset has become the benchmark for testing different methods for analysis of mixture data. See **?** for a recent investigation into the galaxy data in the mixture modelling context.

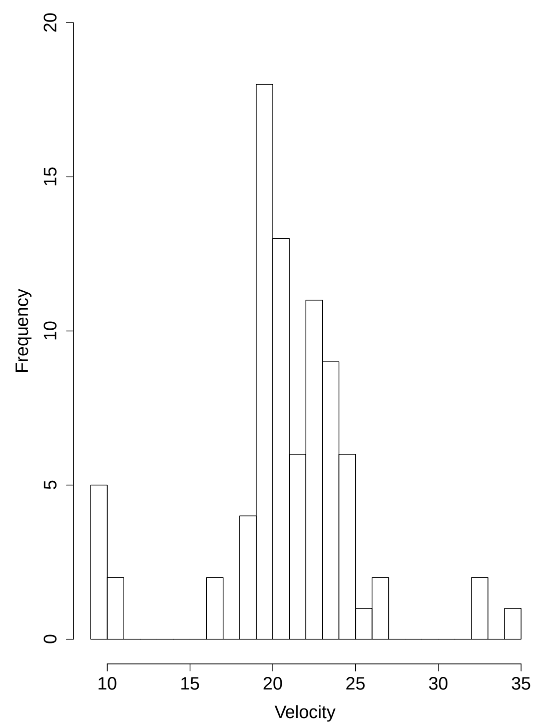

Figure 3.2: Histogram of the velocities of 82 galaxies

An MCMC run on this data spends at least 10% of its iterations in each of $K = 3$, 4 and

5 clusters suggesting that any of these choices could be sensible. We refer to **?** for an interesting summary of the differing posteriors for K achieved using different, but apparently similar, methods on this dataset. Here we will look in detail at the relabelling algorithms applied to the $K = 5$ case. As this is a single data set, it is feasible to use all of the output points from the MCMC for the SEMUP algorithm.

A remarkable stability between the different methods can be found as they recover almost identical values to each other for all the parameters. Table **??** shows an example of these results, the component mean of the fourth component, μ_4. The mean changes between methods, which is due to the difference in dealing with tails of the relabelled posterior by the various methods. Looking at the α-quantiles this is further illustrated by the fact that $q_{0.05}$ and $q_{0.95}$ are rather different between the methods. This suggests that there are adjacent components that are poorly separated. For a parameter in a well-separated component, such as the first component that accounts for the observations in the left-hand peak, almost identical results for the α-quantiles are observed for each relabelling method. Consequently the only major difference between the algorithms can be found in the variance for the estimates of each parameter as the allocation of component estimates from the tails has a large bearing on the estimated variances of the parameters.

Table 3.2: Summary of estimated μ_4 for different relabelling methods across all iterations of the MCMC with $K = 5$. Here, μ_4 is defined as the mean with the fourth smallest ergodic average.

Method	Mean	Posterior quantiles				
		$q_{0.05}$	$q_{0.25}$	$q_{0.50}$	$q_{0.75}$	$q_{0.95}$
IC	23.92	21.81	22.56	23.01	23.58	32.51
PL	22.60	21.33	22.04	22.65	23.09	23.65
CPL	22.39	21.09	21.83	22.43	23.00	23.45
AL	22.49	21.20	21.94	22.55	23.04	23.62
EMP	23.92	21.40	22.09	22.68	23.13	24.13
SEMP	22.60	21.25	22.00	22.63	23.09	24.09
SEMUP	23.37	16.39	22.21	22.88	23.44	34.60

Figure **??** gives the probabilities of the two most likely permutations (calculated from

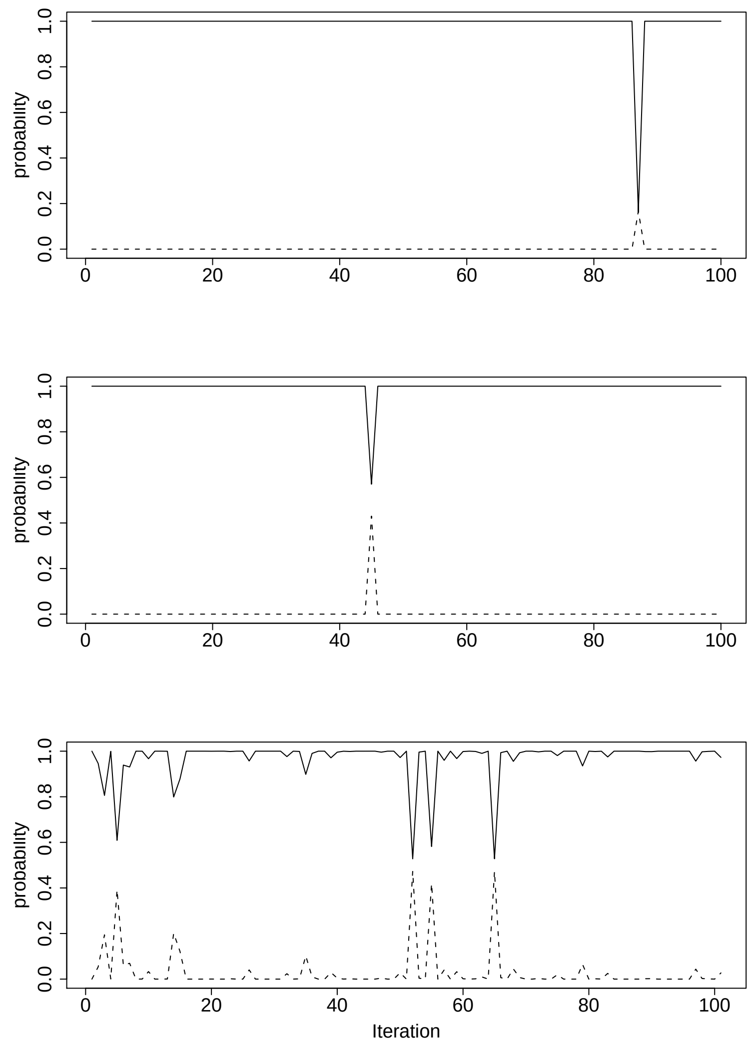

Figure 3.3: Graphs showing the probabilities of the most likely (solid line) and second most likely (dashed line) permutations at 100 iterations of the MCMC sampler for the galaxy data. The three graphs represent models with differing numbers of components — $K = 3$ (top), K=4 (middle) and K=5 (bottom).

Equation **??**) for the galaxy data with the number of components $K = 3, 4, 5$, for 100 thinned iterations in each case. We have used the SEMP relabelling procedure. For $K = 3$ and 4, there is little or no uncertainty over which permutation of the labels is selected. For $K = 5$, however, it is often the case that there are two permutations with reasonable probabilities of being selected. This suggests that there are two components that are virtually indistinguishable, which implies that it may be beneficial to merge them. In this way, there is potential to use this method to help choose the number of components K.

3.3.2 Simulated Data

For a more thorough evaluation of the different relabelling algorithms we now turn to simulated data. We investigated different simulations in which we draw n observations from $\pi N(0,1)+(1-\pi)N(\mu_2,\sigma_2^2)$, for various combinations of (n,π,μ_2,σ_2^2). Each combination is repeated 100 times and the results are averaged over these repeats in order to remove the impact of individual data sets. Since it is computationally not feasible to use the SEMUP algorithm with all available iterations of the MCMC we set the number of posterior points to 100 for use in Equation (**??**). As well as giving estimates of parameters, we give a measure of closeness of the estimated mixture distribution to the true density that we have simulated from, by simulating 10^6 values from the true density and estimating the Kullback–Leibler distance via

$$\varphi = \frac{10^4}{10^6}\sum_{i=1}^{10^6}\log\left\{\frac{f(Y_i;\boldsymbol{\theta}_{\text{true}})}{f(Y_i;\hat{\boldsymbol{\theta}})}\right\},$$

where $\hat{\boldsymbol{\theta}}$ is estimated via the various relabelling methods, and we have rescaled by 10^4 from a usual average to give more readable results.

For situations where the difference between two components is large, that is when either μ_2 was very different from zero or σ_2^2 was very different from 1 (for example, $\sigma_2^2 = 0.1$ or $\sigma_2^2 = 10$), all relabelling algorithms, unsurprisingly, performed well as label switching occurs rarely. We therefore omit the details of these simulations and focus on situations where the two components are very similar. Tables **??**–**??** provide the details of some of the most interesting situations considered.

For the case where $(\pi,\mu_2,\sigma_2^2) = (0.5,2,1)$ and the sample size is varied as $n = 50$ and $n = 100$ (Table **??**), it is immediately striking that for all relabelling algorithms except the IC, the estimates of μ_1 and μ_2 are pushed toward each other with the effect being strongest for the CPL and AL methods, and a moderate effect for the probabilistic strategies. Further, for all relabelling methods, the variances are severely over-estimated and neither feature is improved by an increase sample size, even when raised to $n = 500$ (not shown).

Both of these problems can be attributed to posterior weight on the possibility of both components being in the middle of the dataset with similar means and different variances. This solution, however, yields a rather different interpretation of the components than the one used to generate the data. The high standard deviation of the simulations indicates a high uncertainty in the 'correct' interpretation of the mixture distribution. In terms of the predictive error φ, PL and the probabilistic approaches are best performers in both sample sizes.

When looking at the results for very similar components ($\mu_2 = 0.1$, Table **??**), we see the converse feature of the average estimates of μ_1 and μ_2 being pushed apart from each other. This is caused by the components being virtually indistinguishable so the MCMC responds by moving one component excessively to the left and the other excessively to the right. These results, opposite to the previous case where the components were pushed together, are an illustration of the limitations of using ergodic average estimates for the parameters. For this situation interestingly the CPL and AL method perform better than the other methods, while probabilistic relabelling methods are in the middle. It is also interesting to see that, contrary to the previous set of situations, the estimates of the variance are more or less on target for all algorithms considered. In this case, the predictive error φ is minimized by CPL and AL, although SEMUP performs fairly well.

In Table **??** the components are more distinguishable ($\mu_2 = 2$), but the mixing weights are rather different, with $\pi = 0.1$. In this case, μ_1 and σ_1 are both severely over-estimated while μ_2 and σ_2 are estimated accurately for all relabelling strategies with none of the methods appearing to be superior to the others. Additionally π is also over-estimated strongly which can be attributed to the asymmetry in the posterior distribution. The predictive error φ is smallest for the PL method while it is largest for the IC.

Overall the results indicate that none of the methods compared are performing uniformly

Table 3.3: Average parameter estimates over 100 iterations for different relabelling strategies when $(\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = (0.5, 0, 2, 1, 1)$ for $n = 50$ and $n = 100$. Values in parentheses give the standard deviations of the estimates.

n	θ	IC	PL	CPL	AL	EMP	SEMP	SEMUP
50	μ_1	−0.02 (0.21)	0.19 (0.12)	0.64 (0.19)	0.65 (0.18)	0.20 (0.09)	0.20 (0.16)	0.34 (0.16)
	μ_2	1.90 (0.12)	1.69 (0.03)	1.24 (0.09)	1.23 (0.08)	1.69 (0.01)	1.68 (0.06)	1.54 (0.06)
	σ_1^2	1.63 (0.39)	1.62 (0.31)	1.58 (0.51)	1.58 (0.51)	1.62 (0.37)	1.63 (0.38)	1.66 (0.39)
	σ_2^2	1.61 (0.47)	1.62 (0.55)	1.66 (0.34)	1.66 (0.34)	1.62 (0.48)	1.61 (0.48)	1.58 (0.46)
	π	0.50 (0.06)	0.51 (0.09)	0.49 (0.38)	0.49 (0.39)	0.51 (0.11)	0.46 (0.08)	0.47 (0.11)
	φ	205	97	166	169	96	86	83
100	μ_1	0.07 (0.18)	0.23 (0.23)	0.58 (0.36)	0.58 (0.36)	0.27 (0.28)	0.28 (0.25)	0.30 (0.26)
	μ_2	1.91 (0.19)	1.75 (0.23)	1.39 (0.39)	1.40 (0.38)	1.71 (0.30)	1.70 (0.29)	1.68 (0.27)
	σ_1^2	1.52 (0.39)	1.52 (0.39)	1.50 (0.39)	1.50 (0.39)	1.51 (0.35)	1.52 (0.39)	1.52 (0.36)
	σ_2^2	1.47 (0.36)	1.47 (0.37)	1.49 (0.36)	1.49 (0.36)	1.49 (0.40)	1.47 (0.35)	1.47 (0.39)
	π	0.50 (0.05)	0.50 (0.07)	0.49 (0.24)	0.49 (0.24)	0.50 (0.09)	0.50 (0.09)	0.49 (0.09)
	φ	110	65	188	184	66	67	70

Table 3.4: Average parameter estimates over 100 iterations for different relabelling strategies when $(\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = (0.5, 0, 0.1, 1, 1)$ for $n = 100$. Values in parentheses give the standard deviations of the estimates.

θ	IC	PL	CPL	AL	EMP	SEMP	SEMUP
μ_1	−0.60 (0.24)	−0.42 (0.28)	−0.22 (0.30)	−0.21 (0.30)	−0.47 (0.30)	−0.44 (0.28)	−0.36 (0.29)
μ_2	0.67 (0.22)	0.47 (0.24)	0.27 (0.26)	0.27 (0.26)	0.52 (0.25)	0.50 (0.25)	0.42 (0.25)
σ_1^2	0.95 (0.25)	0.95 (0.31)	0.94 (0.26)	0.94 (0.26)	0.95 (0.24)	0.95 (0.27)	0.96 (0.28)
σ_2^2	0.92 (0.23)	0.91 (0.29)	0.92 (0.23)	0.92 (0.23)	0.92 (0.23)	0.91 (0.24)	0.91 (0.24)
π	0.49 (0.09)	0.49 (0.15)	0.47 (0.29)	0.47 (0.29)	0.49 (0.14)	0.48 (0.14)	0.50 (0.15)
φ	211	37	0.4	0.4	66	50	18

Table 3.5: Average parameter estimates over 100 iterations for different relabelling strategies when $(\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = (0.1, 0, 2, 1, 1)$ for $n = 100$. Values in parentheses give the standard deviations of the estimates.

θ	IC	PL	CPL	AL	EMP	SEMP	SEMUP
μ_1	0.75 (0.40)	0.91 (0.47)	1.07 (0.55)	1.08 (0.55)	0.85 (0.45)	0.91 (0.49)	0.95 (0.49)
μ_2	2.32 (0.20)	2.17 (0.20)	2.00 (0.21)	1.99 (0.21)	2.22 (0.23)	2.16 (0.23)	2.12 (0.20)
σ_1^2	1.39 (0.36)	1.46 (0.46)	1.36 (0.43)	1.35 (0.42)	1.35 (0.36)	1.39 (0.38)	1.38 (0.41)
σ_2^2	1.02 (0.29)	0.95 (0.25)	1.05 (0.26)	1.05 (0.27)	1.05 (0.31)	1.02 (0.29)	1.02 (0.32)
π	0.39 (0.10)	0.36 (0.12)	0.28 (0.18)	0.28 (0.18)	0.38 (0.12)	0.39 (0.13)	0.40 (0.14)
φ	184	51	57	59	125	106	110

better than any of the others leaving the ultimate decision on which method to use to the user. The CPL and AL methods are unstable in terms of the predictive error φ as they perform well when the components are very hard to distinguish, but show poor performance when the components are more separated. Consistent results for φ are obtained for the PL and the probabilistic methods. Based on these results it is, however, evident that the use of ergodic averages can often be detrimental. Due to the large variation in the parameter estimates we believe the SEMUP method is more appropriate as it is probabilistic and moreover avoids conditioning on the parameter estimates. It does, however, depend on the accuracy of the approximation in Equation (**??**) through the value of R.

3.4 Discussion

In this paper we have developed a new class of probabilistic methods for the label switching problem in Bayesian mixture models. The main advantages of these approaches are on the one hand that the tails of the posterior distributions are recovered and on the other hand uncertainty associated with relabelling can be incorporated, features that are not present for deterministic relabelling algorithms. The computation time of the probabilistic methods are either substantially lower than or on par with the existing deterministic methods with the exception of the IC. It is shown through analysis of an observed dataset as well as simulation that the parameter estimates obtained by probabilistic relabelling are virtually the same as for the deterministic approaches suggesting that the above advantages come without any loss.

We also introduce an algorithm for probabilistic relabelling, called SEMUP, that does not rely on ergodic average estimates of parameters as we integrate over a relabelled posterior. Although there is some additional computation required to approximate the relevant integral that also introduced a trade-off between speed and accuracy, the additional time was found to be reasonable for single datasets.

During the evaluation of the methods it was pointed out that some information about the choice of K, the number of components, can be derived from probabilistic relabelling algorithms. Although the full extent of the relevance of probabilistic relabelling for choosing K is still to be evaluated carefully, it does show promise. The uncertainty in the relabelling can be used as an indication that too many components are in the model, since high uncertainty in relabelling suggests that there is ambiguity between adjacent components, implying that it may be better to merge them. Further work will need to be done to get a better understanding of this.

3.5 Appendix

Derivation of Equation (??)

First, $g_r(\nu_r; \hat{\boldsymbol{\gamma}}, \boldsymbol{Y}, \boldsymbol{Z}^{(r)})$ is defined as the probability that permutation ν_r is 'correct', given the data $\boldsymbol{Y}$, the current estimate of the parameters $\hat{\boldsymbol{\gamma}}$, and the allocation vector $\boldsymbol{Z}^{(r)}$, for the r^{th} iteration of the sampler. In an abuse of notation when we write ν_r henceforth we mean 'permutation ν_r is correct'.

Then

$$\Pr[\nu_r|\boldsymbol{Y}, \boldsymbol{Z}^{(r)}, \hat{\boldsymbol{\gamma}}] = \frac{\Pr[\boldsymbol{Y}|\nu_r, \boldsymbol{Z}^{(r)}, \hat{\boldsymbol{\gamma}}]\Pr[\boldsymbol{Z}^{(r)}|\nu_r, \hat{\boldsymbol{\gamma}}]\Pr[\nu_r|\hat{\boldsymbol{\gamma}}]}{\Pr[\boldsymbol{Y}|\hat{\boldsymbol{\gamma}}, \boldsymbol{Z}^{(r)}]\Pr[\boldsymbol{Z}^{(r)}|\hat{\boldsymbol{\gamma}}]}.$$

Now, the terms in the denominator do not depend on ν_r. The permutations ν_r are marginally independent of the parameters $\hat{\boldsymbol{\gamma}}$ — it is exactly this that causes the label switching problem. So we have $\Pr[\nu_r|\hat{\boldsymbol{\gamma}}] = \Pr[\nu_r]$, and we assume that each permutation is equally likely, so we are left with

$$\begin{aligned}
\Pr[\nu_r|\boldsymbol{Y}, \boldsymbol{Z}^{(r)}, \hat{\boldsymbol{\gamma}}] &\propto \Pr[\boldsymbol{Y}|\nu_r, \boldsymbol{Z}^{(r)}, \hat{\boldsymbol{\gamma}}]\Pr[\boldsymbol{Z}^{(r)}|\nu_r, \hat{\boldsymbol{\gamma}}] \\
&= \prod_{i=1}^{n} \Pr[Y_i|\nu_r, Z_i^{(r)}, \hat{\boldsymbol{\gamma}}]\Pr[Z_i^{(r)}|\nu_r, \hat{\boldsymbol{\gamma}}] \\
&= \prod_{k=1}^{K} \prod_{i \in S_k^r} \hat{\pi}_{\nu(k)} f_{\nu(k)} \left(Y_i|\hat{\boldsymbol{\theta}}_{\nu(k)}, \hat{\boldsymbol{\eta}} \right),
\end{aligned}$$

which is the form given in Equation (??).

Chapter 4

Sparsity Methods in High Dimensions

4.1 Introduction

We now switch attention to the second methodological area of the thesis, the recovery of direct effects from data sets in which the number of predictors, p, is larger than the number of observations, n. Obtaining information from a dataset with a large number of predictors is an important issue in statistics. We focus attention on supervised datasets (i.e. those including at least one response variable). High dimensional, supervised datasets commonly arise in all fields of statistics, and are particularly common in genetics. There are at least two kinds of inference one may wish to carry out on such datasets: prediction, and true sparsity recovery (**?**). In prediction, the goal is to estimate or predict the response accurately. True sparsity recovery, on the other hand, focuses attention on the predictors; the goal is to include the 'correct' predictors in the model, where 'correct' refers to those predictors that have a direct effect on the response. A direct effect is a relationship between a predictor and the response that is not explained by any other measured predictors. Those predictors that are associated with the response but do not have a direct effect are said to have indirect effects. There can be many predictors with indirect effects when the predictors are correlated.

The distinction between prediction and true sparsity recovery is now illustrated with an example. Suppose that there are two closely related predictors available for a response. The response may be an indicator of incidence of heart disease, and the predictors body mass index (BMI), and fat percentage (proportion of body mass that consists of fat). Suppose that through an oracle it is known that fat percentage has a direct effect on heart disease in this model, whilst BMI has only an indirect effect, that comes about through its correlation with fat percentage. Assume further that after fitting a model, the result is a regression equation that relates heart disease to BMI but not fat percentage. If the goal was prediction, we would be quite satisfied. The fact that the included predictor is indirectly responsible for heart disease is not important, since we are only concerned with making accurate predictions. If, however, recovery of the true sparsity pattern was the goal, getting the wrong predictor would be more concerning. On the other hand, suppose our fitted model was a regression equation that related heart disease to fat percentage, but the effect size was estimated poorly. This would be a success for recovery of the true sparsity pattern, since the correct predictor is included in the model. The poor estimate of the effect size, however, means that the predictive ability of the model may be poor.

In the remainder of this Chapter we introduce the regression notation, and give an introduction to the lasso (**?**) and related methods. In Chapter **??**, which is the paper **?**, we introduce direct effect testing (DET), which is a novel method that recovers direct effects between binary predictors and a binary response. Chapter **??** consists of the paper **?**, and is a detailed comparison study of how well various methods, including DET, are able to recover true sparsity patterns in various scenarios. Chapter **??** then presents some additional work and extensions that have not been written for publication. We now introduce the notation for this part of the thesis.

Suppose n observations are collected, including a univariate response variable and p predictor variables. Suppose for the moment that there is no missing data. The responses are recorded in a response vector $\boldsymbol{Y} = (Y_1, Y_2, \ldots, Y_n)'$; each predictor variable is recorded in a vector $\boldsymbol{X}_j = (X_{1j}, X_{2j}, \ldots, X_{nj})'$, $j = 1, \ldots, p$, and the predictor variables are combined into an $n \times p$ design matrix $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_p)$. We suppose that there is a linear relationship between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and so consider the standard regression equation

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\delta}, \tag{4.1}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)'$ is a vector of unknown coefficients, and $\boldsymbol{\delta}$ is an $n \times 1$ independent noise vector. For the remainder of this section we assume without loss of generality that the predictors are scaled and centred, i.e. $\sum_{i=1}^{n} X_{ij} = 0$ and $\sum_{i=1}^{n} X_{ij}^2 = 1$. Additionally without loss of generality, for each j we assume $\sum_{i=1}^{n} Y_i = 0$.

We will suppose that the true solution to Equation (**??**) is sparse, meaning that $\beta_j = 0$ for many $j = 1, \ldots, p$. Let $D = \{j : \beta_j \neq 0\}$, so that D denotes the set of predictors that should be included in the model, i.e. the 'true sparsity pattern'. Let $|D|$ denote the number of elements in the set D. It is typically assumed that most of the predictors turn out to be excluded from the model, i.e. $|D| = s \ll p$ (see, for example, **?**).

We are interested in the case where p is large, even $p \gg n$. The classical solutions to the problem of large p fall into at least three families: principal component regression (**?**), ridge regression (**?**) and subset selection (for example, **?**). The principal component regression family includes methods such as partial least squares (**?**), and is still an area of active research, with recent developments including sparse sufficient data reduction (**?**). Since principal component regression is focussed on prediction rather than sparsity, we do not consider it further here. Ridge regression and subset selection, which are prediction and sparsity methods respectively, are combined by the lasso (**?**).

4.2 The Lasso

The lasso (least absolute shrinkage and selection operator) was popularised by the seminal paper of **?**. It is also known as basis pursuit in the machine learning and wavelets literature (**?**). The lasso can be described as a *penalised regression* method, placing it in the same class as ridge regression (**?**).

Recall that the ordinary least squares (OLS) solution to the regression equation (**??**) is obtained by minimising the residual sum of squares

$$\hat{\boldsymbol{\beta}} == \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^2.$$

In penalised regression, a penalty function $p(\cdot)$ is added, with a tuning parameter λ, to give

$$\hat{\boldsymbol{\beta}} == \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \, (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^2 + \lambda p(\boldsymbol{\beta}). \tag{4.2}$$

The ridge regression solution is obtained by setting $p(\boldsymbol{\beta}) = \sum_{j=1}^{p} \beta_j^2$, and has the property that the coefficients β_j, $j = 1, \ldots, p$ are shrunk towards zero. The amount of shrinkage that occurs is controlled by the size of the tuning parameter λ. From a Bayesian perspective, ridge regression is equivalent to assigning independent normal priors to the β_j's. The method does not cause sparsity. Indeed, $\beta_j \neq 0$ almost surely for all j. Generalisation of ridge regression to the penalty $p(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|^{\gamma}$, for $\gamma \geq 0$ — called bridge regression — is considered by **?**.

The lasso uses $p(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|$, corresponding to a bridge regression with $\gamma = 1$. The consequence of using a linear penalty, rather than the quadratic penalty of ridge regression, is that sparsity is achieved, since some of the coefficients β_j will be zero. This is due to the non-differentiability of $p(\boldsymbol{\beta})$ at zero. From a Bayesian perspective, lasso regression is equivalent to assigning independent double exponential priors to the β_j's. The lasso is also commonly expressed in the following form:

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^2 \text{ subject to } \sum_{j=1}^{p} |\beta_j| < t, \tag{4.3}$$

where t now takes the role of the tuning parameter. The forms of Equations (**??**) and (**??**) are equivalent, in the sense that for each t there exists a λ that gives the same solution, and vice-versa.

The applicability of the lasso was further increased with the introduction of the LARS (least angle regression) algorithm (**?**). This algorithm can be used to reconstruct the entire lasso path (i.e. the value of each β_j, $j = 1, \ldots, p$, as a *continuous* function of the penalty λ) in the time it takes to carry out OLS regression. Not only is LARS a useful calculation algorithm, its method of operation is also instructive in understanding how the lasso behaves, and its relation to other methods. Therefore, we now consider the LARS algorithm in some detail. The algorithm proceeds as follows, where $\text{cor}(\cdot,\cdot)$ denotes correlation.

1. Start with the null model: $\beta_j = 0$ for all $j = 1, \ldots, p$.

2. Enter the variable $\boldsymbol{X}_j$ with the largest absolute value of $\text{cor}(\boldsymbol{X}_j, \boldsymbol{Y})$ into the model.

3. Gradually adjust β_j so that $\text{cor}(\boldsymbol{X}_j, \boldsymbol{Y} - \boldsymbol{X\beta})$ decreases, until there exists a variable $\boldsymbol{X}_k$ with an equal residual correlation, $\text{cor}(\boldsymbol{X}_k, \boldsymbol{Y} - \boldsymbol{X\beta}) = \text{cor}(\boldsymbol{X}_j, \boldsymbol{Y} - \boldsymbol{X\beta})$.

4. Enter variable $\boldsymbol{X}_k$ into the model, and now adjust the coefficients β_j and β_k such that $\text{cor}(\boldsymbol{X}_k, \boldsymbol{Y} - \boldsymbol{X\beta})$ and $\text{cor}(\boldsymbol{X}_j, \boldsymbol{Y} - \boldsymbol{X\beta})$ are both decreasing at the same rate (so they remain equal).

5. Continue adjusting until there exists a variable $\boldsymbol{X}_l$ with an equal residual correlation, $\text{cor}(\boldsymbol{X}_l, \boldsymbol{Y} - \boldsymbol{X\beta}) = \text{cor}(\boldsymbol{X}_k, \boldsymbol{Y} - \boldsymbol{X\beta}) = \text{cor}(\boldsymbol{X}_j, \boldsymbol{Y} - \boldsymbol{X\beta})$,...

This process continues in the obvious way, with the condition that if any coefficient β_j becomes zero at any point, the variable $\boldsymbol{X}_j$ is removed from the model at that point (but may be re-introduced later). The process terminates when either the OLS solution is reached, when there are as many active variables in the model as there are observations, or when some other criterion specified by the user is reached.

An artificial example with three variables, illustrating the LARS procedure, is given in Figure **??**. The y-axis represents the standardised values of the coefficients, while the x-axis represents the severity of the sparsity penalty, which depends on λ (becoming less severe from left to right). The first variable to enter the model is $\boldsymbol{X}_2$, at the point labelled '1' (at the top of Figure **??**). Then at the point labelled '2', variable $\boldsymbol{X}_1$ also enters; notice that at this point the path of the coefficient of $\boldsymbol{X}_2$ adjusts its direction. Finally variable $\boldsymbol{X}_3$ enters the model at point '3', and the OLS solution is reached at the right hand end of the Figure, and the procedure terminates.

It is often overlooked that, for $p > n$, the lasso estimator is multimodal (see, for example **?**). This is an important issue that needs further work and consideration. In this work, however, for simplicity we identified a single solution only for each lasso regression.

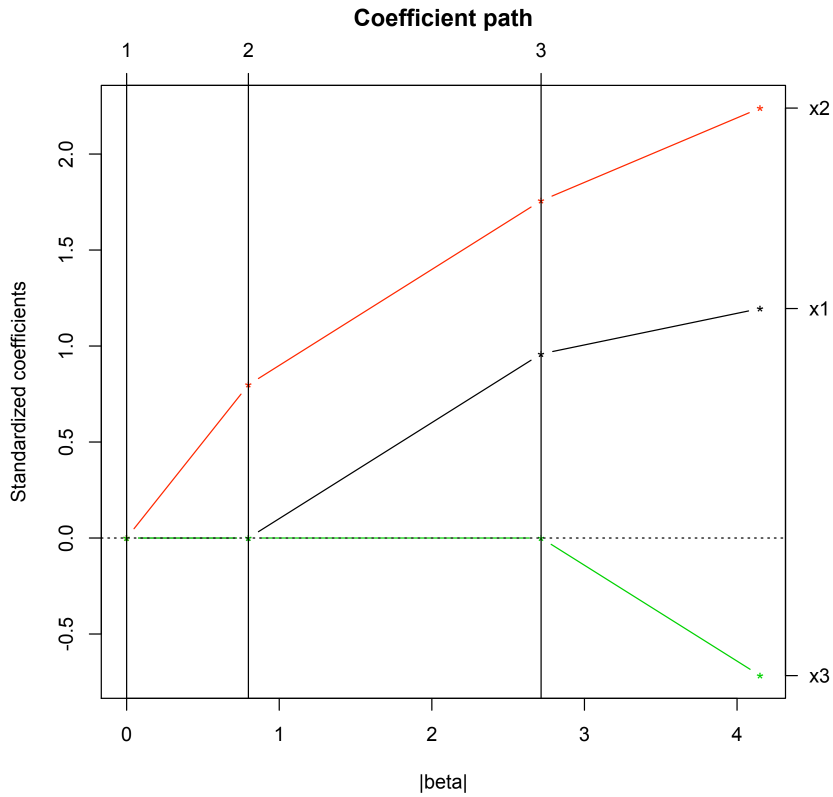

Figure 4.1: Illustrative example of the LARS procedure with three variables

4.2.1 Choosing the Tuning Parameter λ

Choosing the tuning parameter λ, or equivalently selecting the point on the LARS path to take as the solution, is a model selection problem. A popular method to select the tuning parameter λ is k-fold cross validation. This is introduced in **?** and specifically applied to the lasso in **?**. In brief, the idea is to split the observations into k equally sized groups ($k = 10$ is a common choice for the lasso), and for each k^{th} group, use the remaining $k-1$ groups to fit the lasso, and calculate the error in predicting the k^{th} group of observations. The average prediction error can then be calculated over the k parts. In the case of the lasso, the value of λ that minimises the average prediction error would be chosen. Standard cross-validation techniques involve splitting the data into two groups, fitting the various models with one group, then selecting the model that minimises the prediction error of the second group. The main advantage of k-fold cross validation, over the two groups method, is the increase in power gained by using more observations to fit the models (see, for example, **??**).

An alternative, less computationally intensive method to choose λ is to use standard model fit criteria such as Akaike's information criterion (AIC) (**?**) and Schwarz's information criterion (BIC) (**?**). The AIC is essentially a prediction error criterion, where the 'best' model is obtained by minimising

$$\mathrm{AIC} = 2p_a - 2\log L,$$

where p_a is the number of active (nonzero) coefficients in the model, and L is the likelihood evaluated at the corresponding values of the parameters. The focus of the BIC, on the other hand, is the recovery of the true model, and one selects the model minimising

$$\mathrm{BIC} = p_a \log n - 2\log L,$$

where n represents the number of observations and the other parameters are defined as above. The BIC tends to select solutions with less coefficients than the AIC.

We have preferred to use model fit criteria than cross-validation in this work, because of the computational savings. These savings has been essential especially for the extensive simulations carried out in Chapter **??** (**?**). Also, since we focus attention on true sparsity recovery rather than prediction, we see BIC as a more appropriate choice than both AIC and cross validation. Further comments on the choice of the tuning parameter λ are made throughout this work.

4.3 Extensions and Alternatives to the Lasso

The importance and influence of the lasso is reflected in a vast array of literature devoted to its extensions and alternatives, following the original paper of **?**. Therefore, the discussion in this section is necessarily an incomplete summary. We focus on the methods that are considered in the subsequent Chapters.

4.3.1 Screen and Clean

A major criticism of the lasso is that it is very difficult to assign significance to predictors. A solution to this is given by the 'screen and clean' method (**?**). The idea of the procedure is to split the data into two groups, $\mathcal{D}_1$ and $\mathcal{D}_2$, of equal size $n/2$.

Group $\mathcal{D}_1$ is used for the 'screen' stage. This involves fitting a dimension reduction method, such as lasso. **?** discuss splitting the data into three groups rather than two, and using the second group for cross-validation to select the tuning parameter λ. Whilst this is necessary for the theoretical developments of the method, simulations demonstrate that a two group approach and selecting λ by k-fold cross validation is superior. In this thesis, we have used BIC to select λ when using screen and clean, and have hence used the two group approach. After the optimal tuning parameter λ has been chosen, the predictors $\{\boldsymbol{X}_j\}$ with $\hat{\beta}_j = 0$ are discarded from the model — only the predictors with nonzero coefficients are retained, and carried forward to the second stage.

The second stage (the 'clean' stage) of the procedure involves fitting the second half of the data, $\mathcal{D}_2$, using only the retained predictors. **?** propose that this is done by fitting an OLS regression, and declaring significance of the remaining predictors according to their p-values. These p-values are subject to a Bonferroni correction (**?**), according to the number of predictors p_a remaining in the model.

A weakness of the screen and clean method is the sensitivity of the p-values to the selection of $\mathcal{D}_1$ and $\mathcal{D}_2$. **?** remedy this by proposing a 'multiple split' procedure, in which screen and clean is carried out multiple times, B say, with new groups $\mathcal{D}_1^b$ and $\mathcal{D}_2^b$ chosen for each $b = 1, \ldots, B$.

4.3.2 Stability Selection

Stability selection (**?**) tackles the problem of significance testing for predictors by resampling. The idea is to select predictors that are 'stable', in the sense that their coefficients are nonzero in a certain proportion of lasso regressions carried out on resampled copies of the data. Each sample of the data is generated by selecting $n/2$ observations at random (without replacement), and repeating this process B times, say. This is the method used by

?, see ? for discussion of the choice $n/2$. Lasso regression is then fitted to each sample of the data, with an optimal tuning parameter $\hat{\lambda}(b)$ chosen for each resampled copy, $b = 1, \ldots, B$. The predictors with nonzero coefficients are recorded from each sample of the data. Let m_j be the count of times the j^{th} predictor is nonzero. We then calculate, for each predictor $\boldsymbol{X}_j$,

$$\pi_j = m_j/B,$$

the proportion of times that each predictor is present in the fitted model over the B resamplings. Predictor $\boldsymbol{X}_j$ is then deemed to be significant if $\pi_j > \pi_{\text{thr}}$, where π_{thr} is a tuning parameter. Stability selection does not provide in itself a method to estimate effect sizes, but one could, for example, fit a linear regression on the variables that have been selected.

A strength of stability selection is that the predictors recovered are insensitive to the choice of π_{thr} for $\pi_{\text{thr}} \in (0.6, 0.9)$, and insensitive to the tuning parameter λ used in each of the lasso regressions (?), provided a reasonable value of λ is chosen. The procedure is also proven, in the case of exchangeable coefficients, to bound the expected number of incorrectly detected predictors. Even when the exchangeability conditions are violated, the false positive rate is shown to be well controlled (?, and see also Chapter ?? in this work). On the other hand, the nature of stability selection means that it will fail in the presence of highly correlated predictors. Consider a set of, say, 10 similar predictors, one of which has a true large effect on the response. Then each predictor may have $\pi_j \approx 1/10$, as the lasso regression tends to select one of the predictors 'at random' to represent the effect (see, for example, ?). This proportion π_j would be smaller than any reasonable threshold π_{thr}, meaning that none of the 10 predictors would be declared significant, and the model would fail to detect the true large effect.

4.3.3 The Dantzig Selector

The Dantzig selector is developed by ?, and is quite distinct from the lasso, rather than being an extension to it. Whereas the lasso penalty is a special case of Equation (??), the Dantzig selector is quite different. The minimization of the residual sum of squares is replaced by the minimization of the infinity norm of the *correlated* residuals:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \; \|\boldsymbol{X}'(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\|_\infty + \lambda\|\boldsymbol{\beta}\|_1. \tag{4.4}$$

The infinity norm, also known as the sup-norm, controls the value of the largest element of the vector,

$$\|(c_1, c_2, \ldots, c_m)'\|_\infty = \sup_i \{c_1, c_2, \ldots, c_m\}.$$

The reason for the development of the Dantzig selector was that theoretical results were more readily available, and a suggested shift of emphasis from prediction to sparsity recovery (**?**). Recent work, however, has established similar properties for the lasso (for example, **?**). **?** have developed an algorithm that allows the entire path of the Dantzig selector to be calculated, analogous to the LARS algorithm for lasso. The work also shows that in sparse situations, the Dantzig and lasso solutions are identical, and hence the Dantzig selector has not been considered in Chapters **??** or **??**. Instead, the Dantzig selector is considered in Chapter **??**, where we highlight similarities of the direct effect testing method to the Dantzig selector.

Chapter 5

Direct Effect Testing: A Two-Stage Procedure to Test for Effect Size and Variable Importance for Correlated Binary Predictors and a Binary Response

Abstract

In applications such as medical statistics and genetics, we encounter situations where a large number of highly correlated predictors explain a response. For example, the response may be a disease indicator and the predictors may be treatment indicators or single nucleotide polymorphisms (SNPs). Constructing a good predictive model in such cases is well studied. Less well understood is how to recover the 'true sparsity pattern', that is finding which predictors have direct effects on the response, and indicating the statistical significance of the results. Restricting attention to binary predictors and response, we study the recovery of the true sparsity pattern using a two-stage method that separates establishing the presence of effects from inferring their exact relationship with the predictors. Simulations and a real

data application demonstrate the method discriminates well between associations and direct effects. Comparisons with lasso based methods demonstrate favourable performance of the proposed method.
Keywords: Contingency table; Direct effect; High dimensional; Lasso; Noncentral hypergeometric distribution; Sparsity.

5.1 Introduction

It is commonplace in applications of statistics to encounter situations in which a large number of predictors are available to explain a response. Consider the standard regression

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{5.1}$$

where $\boldsymbol{Y}$ is an $n \times 1$ response vector explained by an $n \times p$ design matrix $\boldsymbol{X}$ through an unknown $p \times 1$ coefficient vector $\boldsymbol{\beta}$ with $n \times 1$ noise vector $\boldsymbol{\varepsilon}$. Having a large number of predictors, p, possibly even $p > n$, should intuitively be beneficial, as we are maximising the information available to explain the response. From the perspective of producing a good predictive model, this is true, and many methods are available for this objective, such as principal component regression (**?**), partial least squares (**?**), ridge regression (**?**) and more recent methods such as sparse sufficient data reduction (**?**).

In this paper, however, our focus is in recovering the so-called 'true sparsity pattern' (**?**), in which we search for a subset of predictors deemed to have a 'direct effect' on the response, that is an effect that is attributed to the predictor in question rather than being due to the correlation of the predictor with other important predictors. A more formal definition is given later. We wish to find a sparse solution to the regression given in Equation (**??**) and in particular carry out significance tests of variable importance. The lasso (**?**) is a very popular sparse estimator, where sparsity is induced by applying an L_1 penalty to the size of the vector $\boldsymbol{\beta}$. It is computationally fast thanks to the least angle regression algorithm (LARS, **?**). Other possibilities for sparse estimation include subset selection (**?**), the Dantzig selector (**?**) and sure independence screening (**?**). For the lasso, much work has been carried out concerning consistency in terms of sparse pattern recovery (see, for example, **???**).

Until recently, it has not been possible to reliably ascertain significance of parameters included in a sparse model, that is to test for variable importance. Although standard errors of lasso parameters are available (**??**) these are difficult to interpret because of the discontinuity of the sampling distribution of the parameters. In the situation where the predictors in the model are not too highly correlated, recent methods that address the problem with significance testing include the 'screen and clean' method (**??**), and stability selection (**?**). Such methods are also appropriate when, in the highly correlated predictor case, it is satisfactory to recover predictors that are correlated with the true origins of the effects. Carrying out significance tests in the presence of multicollinearity is, however, according to Meinshausen (**?**, p. 266) 'in some sense ill-posed'.

There are many situations, however, where multicollinearity is present and we are nevertheless interested in recovering the true sparsity pattern, along with ascertaining the significance of our result. For example, in genomewide association studies we study a number of sites on the genome called single nucleotide polymorphisms (SNPs) which are highly correlated with each other. We would like to identify exact regions on the genome that influence the risk of disease, so that appropriate interventions can be considered. Typically, a large number of SNPs are not measured, so true sparsity only corresponds to improving localisation of the effect, not identifying the true causal SNP. The problem of multicollinearity can be seen by considering a group $\mathcal{J}$ of highly correlated predictors, one of which has a true non-zero regression coefficient (or direct effect). In such a situation the lasso will select one variable from $\mathcal{J}$, but there is no stability in which variable is selected. This is noted in **?**, where the 'elastic net' is proposed as a solution, which modifies the lasso by adding an L_2 penalty, promoting inclusion of all the predictors in the group $\mathcal{J}$. Whilst this improves the sensitivity of recovering the sparsity pattern, this is at the expense of inclusion of a potentially large number of unrelated predictors in the model. Additionally, effect sizes become difficult to interpret as they are shared amongst the correlated predictors. Such an approach is useful, for example, in the recovery of gene networks, but not for the true sparsity recovery problem considered here. Meinshausen **?** adopts a hierarchical approach, in which he looks for significance at the level of groups of variables, rather than the level of individual variables. This is sensible, since in the case of the group $\mathcal{J}$ of highly correlated predictors, it can be easy to identify that at least one member of the group has a direct effect, but very difficult to identify which member(s) of the group have the effect. The method, however, relies upon the selection of an appropriate hierarchical clustering regime, and it is apparent that the results will depend upon the clustering method chosen.

In this paper we introduce an alternative two-stage method that allows separation of the two inherent kinds of uncertainty: presence of an effect that is sufficiently large to be deemed significant and which predictor(s) the effect is allied to. The application of the method is to 'fine mapping' problems, where the correlation is particularly high and may violate the standard correlation structure assumptions relied upon by other methods for consistency results (see **?**, for a summary of these assumptions and further references). Consequently, our method makes no claims about consistency of variable selection. Instead, the idea is to acknowledge uncertainty about which predictor is the source of a given effect by providing probabilities that a direct effect arises from each of a collection of predictors. Currently, we restrict attention to binary predictors and response. As the method identifies direct effects, we will call it direct effect testing (DET).

In the remainder of the manuscript, we formally define the methodology in Section **??**, before we investigate its behaviour on simulated data and real data (in Section **??** and Section **??**). We conclude with a brief summary and discussion in Section **??**.

5.2 Method

5.2.1 Correlated Lasso

Suppose we are interested in a response $\boldsymbol{Y}$, and its relationship to a set of p predictors $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p)$. Consider the situation where we have n complete observations; suppose that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are scaled and centred, so that $\sum_{i=1}^n x_{ij} = 0$ and $\sum_{i=1}^n x_{ij}^2 = 1$ for each $j = 1, \ldots, p$, and $\sum_{i=1}^n y_i = 0$ and $\sum_{i=1}^n y_i^2 = 1$. We are interested in finding direct effects between the predictors and the response. Different definitions of the term 'direct effect' appear in the literature. For example, **?** defines a direct effect as a dependence between two variables that is not mediated by a third variable. In this paper, we define a direct effect to be a partial dependence between the response $\boldsymbol{Y}$ and a covariate of interest $\boldsymbol{X}_j$. Formally, this can be expressed as the complement of a conditional independence statement about $\boldsymbol{X}_j$ and $\boldsymbol{Y}$,

$$(\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X}_j | \boldsymbol{X}_{-j})^c.$$

where $\boldsymbol{X}_{-j}$ denotes all predictors except $\boldsymbol{X}_j$. Alternatively, in a graphical model this would correspond to an edge between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$. In a regression of $\boldsymbol{Y}$ on all the predictors $\boldsymbol{X}$, it corresponds to $\beta_j \neq 0$. This is a more general definition than that of **?**, since the relationship between the predictors $\boldsymbol{X}$ is not specified — there can be confounders, effect modifiers, or overlap in information between the predictors. The application we have in mind is where the predictors are genetic markers, where the relationship is induced by spatial correlations that occur as a consequence of the inheritance.

Direct effects can be difficult to identify when the predictors are highly correlated. To see this, suppose that there is a direct effect between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$, and a further covariate $\boldsymbol{X}_k$ is highly correlated with $\boldsymbol{X}_j$. Then a correlation is induced between the covariate $\boldsymbol{X}_k$ and the response $\boldsymbol{Y}$, but this does not mean there is a direct effect between $\boldsymbol{X}_k$ and $\boldsymbol{Y}$. We will call the resulting association between $\boldsymbol{X}_k$ and $\boldsymbol{Y}$ an indirect effect. Our goal is then to identify a small number of direct effects from a potentially much larger number of associations.

To proceed, consider multiplying both sides of the regression equation (Equation **??**) by the transposed design matrix $\boldsymbol{X}'$ and the reciprocal of the number of observations, n^{-1}, giving the normal equations

$$\boldsymbol{R}^y = \boldsymbol{R}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{5.2}$$

where $\boldsymbol{R} = n^{-1}\boldsymbol{X}'\boldsymbol{X}$ is the empirical correlation matrix of $\boldsymbol{X}$, and $\boldsymbol{R}^y = n^{-1}\boldsymbol{X}'\boldsymbol{Y}$ is the correlation vector whose j^{th} entry is the empirical correlation of predictor $\boldsymbol{X}_j$ with the response $\boldsymbol{Y}$. The vector $\boldsymbol{\epsilon} = \boldsymbol{X}'\boldsymbol{\varepsilon}$ is the correlated error term. Clearly, when $\boldsymbol{R}$ is invertible, the equation leads to the standard least squares estimate, $\hat{\boldsymbol{\beta}}_{\text{ls}}$. In general, when p is large, it is desirable to control the estimate by introducing some regulation. For example, one could solve Equation (**??**) using an L_1 constraint on the coefficients, $\boldsymbol{\beta}$, known as the lasso (**?**).

This can be written as

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

We consider instead applying the lasso constraint to Equation (**??**), which corresponds to

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\boldsymbol{X}'(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{5.3}$$

In fact, if one replaces the L_2 norm on the correlated residuals with the L_∞ norm in Equation (**??**), this would correspond to the Dantzig selector (**?**).

The advantage we exploit in this paper is that under the proposed formulation in Equation (**??**), the distribution of the correlated error term $\boldsymbol{\epsilon}$ can be exactly determined, for binary variables at least. Moreover, it does not depend on the uncertainty in the estimation of $\boldsymbol{\beta}$ that is a result of multicollinearity.

5.2.2 Special Case: Binary Variables

In this paper we focus on the special case where the predictors and the response are binary. The overall approach for binary variables can be summarised as follows; detailed descriptions of each steps follow:

1. Construct a regression equation based on the empirical correlations between the predictors and response. Fit this regression using a lasso regression, carrying out constrained minimisation of the correlated residuals rather than the raw residuals (Equation **??**).

2. Carry out significance testing by separating the uncertainty into two levels,

Stage I: Carry out significance testing on effect sizes using Fisher's noncentral hypergeometric as the null distribution (Section **??**).

Stage II: Account for uncertainty in the origin of the effect using counting rules (Section **??**).

Let $\boldsymbol{Y}^u$ and $\boldsymbol{X}_j^u$ denote unstandardised versions of the binary variables $\boldsymbol{Y}$ and $\boldsymbol{X}_j$, taking values 0 and 1. Then, Table **??** gives notation for the 2×2 Table. Without loss of generality we assume in the sequel that the correlation of each predictor $\boldsymbol{X}_j$, $j = 1, \ldots, p$, with the response $\boldsymbol{Y}$ is non-negative, reversing the binary coding for $\boldsymbol{X}_j$ whenever this does not hold.

Table 5.1: Notation for a 2×2 contingency table for a binary response $\boldsymbol{Y}^u$ and binary predictor $\boldsymbol{X}_j^u$

	Observed Counts		
	$\boldsymbol{Y}^u = 0$	$\boldsymbol{Y}^u = 1$	Total
$\boldsymbol{X}_j^u = 0$	a_j	b_j	t_{0j}
$\boldsymbol{X}_j^u = 1$	c_j	d_j	t_{1j}
Total	s	r	n

Under the null hypothesis of no association between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$, the count a_j in the contingency table (Table **??**) is distributed according to a hypergeometric distribution with mean μ_{0j} and variance σ_{0j}^2, given by

$$\begin{aligned} \mu_{0j} =& \frac{st_{0j}}{n}, \\ \sigma_{0j}^2 =& \frac{rst_{0j}t_{1j}}{n^2(n-1)}. \end{aligned}$$

Writing $z_j = \sigma_{0j}^{-1}(a_j - \mu_{0j})$, it can be seen that (see Appendix **??**),

$$z_j = \sqrt{n}\hat{\rho}_{y,j}, \tag{5.4}$$

where $\hat{\rho}_{y,j}$ is the empirical correlation between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$. Therefore we can relate back to Equation (**??**), so that for each z_j,

$$z_j = \sum \beta_k n^{-1/2} \hat{\rho}_{j,k} + \epsilon_j, \tag{5.5}$$

where $\hat{\rho}_{j,k}$ is the empirical correlation between $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$.

We are interested, however, in direct effects rather than associations. A hypothesis test of a direct effect is

$$\begin{aligned} \tilde{H}_0^j &: \quad \boldsymbol{X}_j \text{ is not directly affecting } \boldsymbol{Y}, \\ \tilde{H}_1^j &: \quad \boldsymbol{X}_j \text{ is directly affecting } \boldsymbol{Y}. \end{aligned}$$

Regardless of which of the above hypotheses apply, the count a_j is distributed according to Fisher's noncentral hypergeometric distribution (**?**) with, say, mean $\tilde{\mu}_{\omega j}$ and variance $\tilde{\sigma}^2_{\omega j}$, under $\tilde{H}^j_\omega$, $\omega = 0, 1$. Under $\tilde{H}^j_1$ the noncentrality of the distribution is allowed to include a potential direct effect between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$, but under $\tilde{H}^j_0$ the noncentrality accounts for indirect effects only. Once the mean, $\tilde{\mu}_{\omega j}$ is known, the variance can be approximated (**?**) as:

$$
\begin{aligned}
\tilde{\sigma}^2_{\omega j} &\approx \frac{ngh}{(n-1)(t_{0j}h + t_{1j}g)}, \\
g &= \tilde{\mu}_{\omega j}(t_{0j} - \tilde{\mu}_{\omega j}), \quad h = (s - \tilde{\mu}_{\omega j})(\tilde{\mu}_{\omega j} + t_{1j} - s).
\end{aligned} \tag{5.6}
$$

As each β_k in Equation (**??**), denotes the direct effect between predictor $\boldsymbol{X}_k$ and $\boldsymbol{Y}$, we expect most of these to be zero, and $\beta_k \neq 0$ means that predictor $\boldsymbol{X}_k$ has a direct effect on the response Y. Since all β_k, $k = 1, \ldots, p$, are unknown, we will estimate them as $\hat{\beta}_k$, $k = 1, \ldots, p$ via the approach of Equation (**??**), using the least angle regression algorithm (**?**). In order to choose the constraint on the lasso, note that if we take $E(\epsilon_j) = 0$ for each j,

$$
\sum_{j=1}^{p} \mathrm{var}(\epsilon_j) = E\left(\epsilon_j^2\right) = \sum_{j=1}^{p} \frac{\tilde{\sigma}^2_{1j}}{\sigma^2_{0j}}. \tag{5.7}
$$

We therefore select the point on the lasso path where $\sum_{j=1}^{p} \epsilon_j^2$ is equal to its expectation (Equation **??**). The noncentral variance $\tilde{\sigma}^2_{1j}$ depends upon the current noncentrality estimate, hence is recalculated for every step along the lasso path.

The model (**??**) is not homoskedastic because $\mathrm{var}(\epsilon_j) = \sigma_j^2/\sigma^2_{0j}$, so the variances depend on the size of the noncentrality of each predictor $\boldsymbol{X}_j$. However, scaling by the standard deviation under each H^j_0 provides some stability. Furthermore, the more severe the noncentrality of $\boldsymbol{X}_j$, the smaller its variance tends to be, so there will not be points that exert excessive leverage on the linear model due to large variances. Note further that the ϵ_js are not independent as they are correlated residuals. Standard regression carried out in a situation of non-independent errors, however, leads to coefficient estimates that are still unbiased, but are unlikely to be the best linear unbiased estimator. In this framework, the variance of the estimators is large in the presence of multicollinearity in the predictors. This is taken care of in the two-stage procedure outlined below.

Ideally, we would like to carry out the hypothesis tests $(\tilde{H}_0^j, \tilde{H}_1^j)$ to establish whether or not a direct effect exists between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$, for each $j = 1, \ldots, p$. For a given j, if we knew the direct effects on the other predictors, $\boldsymbol{\beta}_{-j}$, we could calculate the indirect effect between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$, and hence the noncentrality of the noncentral hypergeometric null distribution. Then, the distribution of a_j under $\tilde{H}_0^j$ would have mean $\tilde{\mu}_{0j}$ and variance $\tilde{\sigma}_{0j}^2$, where it is natural to express the mean as

$$\tilde{\mu}_{0j} = \mu_{0j} + \sigma_{0j} \sum_{k \neq j} \beta_k n^{-1/2} \hat{\rho}_{j,k}.$$

This comes about by taking the central mean μ_{0j}, and estimating the null noncentrality parameter as a linear combination of all the indirect effects between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$, and then $\tilde{\sigma}_{0j}^2$ is estimated via Equation (**??**). Thus any remaining association can be attributed to a direct effect.

Unfortunately, we only have an estimate, $\hat{\boldsymbol{\beta}}$, and hence we cannot carry out the above hypothesis tests explicitly. We therefore resort to a two-stage procedure in which we separate the uncertainty in $\hat{\boldsymbol{\beta}}$ into effect size uncertainty and predictor assignment uncertainty.

5.2.3 DET Stage I — Hypothesis Testing for Effect Size

For a set $\mathcal{J}$ of highly correlated predictors, where at least one has a direct effect, the lasso will select one variable from the group (**?**). Therefore, the coefficient estimate $\hat{\beta}_j$ assigned to predictor $\boldsymbol{X}_j$ can be used to estimate the *size* of the corresponding effect, but we must bear in mind that $\boldsymbol{X}_j$ may not be the actual predictor from which the effect originates. We test for significance of the size of the effect assigned by the lasso to each predictor using a Fisher's noncentral hypergeometric null distribution with the estimate $\hat{\boldsymbol{\beta}}$ used to determine the non-centrality. Denoting the resulting mean by $\hat{\tilde{\mu}}_{0j}$ and the variance by $\hat{\tilde{\sigma}}_{0j}^2$ we obtain,

$$\hat{\tilde{\mu}}_{0j} = \mu_{0j} + \sigma_{0j} \sum_{k \neq j} \hat{\beta}_k n^{-1/2} \hat{\rho}_{j,k},$$

and again we can estimate the variance via Equation (**??**). The test statistic is then calculated as

$$T = \frac{z_j - \hat{\hat{\mu}}_{0j}}{\hat{\hat{\sigma}}_{0j}},$$

and this can either be tested against the relevant non-central hypergeometric distribution, or provided the margins of the contingency table are sufficiently large, an approximation to a standard normal distribution is possible.

Since we are carrying out p significance tests, a multiple testing correction is needed. We have used a Bonferroni correction in the results and simulations for this paper for transparency. Any other multiple testing method would offer an improvement over this; which method is appropriate would depend on the application.

5.2.4 DET Stage II — Uncertainty in Direct Effect Predictor Assignment

Suppose predictor $\boldsymbol{X}_j$ has a direct effect on the response $\boldsymbol{Y}$, but is highly correlated with predictor $\boldsymbol{X}_k$. Then by chance it may happen that $\hat{\rho}_{y,k} > \hat{\rho}_{y,j}$, and thus the lasso wrongly identifies the effect on predictor $\boldsymbol{X}_k$ (see also **?**). For each detected effect, we therefore identify a class of predictors from which each effect could truly have originated. Moreover, we allocate a probability to each predictor in this class measuring the likelihood that the effect originated from that predictor. Returning to the graph theory analogy, in the first stage we have established the number of edges originating from the response $\boldsymbol{Y}$, and roughly where each edge leads. We now acknowledge uncertainty, over a set of vertices, for each edge.

When an effect is declared on a predictor $\boldsymbol{X}_k$ in stage I, we generate a set $\{\boldsymbol{X}_j : j \in \mathcal{J}\}$ of predictors highly correlated with $\boldsymbol{X}_k$ (including $\boldsymbol{X}_k$ itself). Then for each $j \in \mathcal{J}$ we would like to calculate $p_{j|k} = \mathrm{pr}(\boldsymbol{X}_j \text{ true direct effect}|\boldsymbol{X}_k \text{ declared direct effect})$.

To proceed we use the result that

$$\begin{aligned} p_{j|k} \quad \propto \quad & \frac{\mathrm{pr}(\boldsymbol{X}_k \text{ declared DE}|\boldsymbol{X}_j \text{ true DE}, \boldsymbol{X}_j \text{ or } \boldsymbol{X}_k \text{ declared DE})}{1 - \mathrm{pr}(\boldsymbol{X}_k \text{ declared DE}|\boldsymbol{X}_j \text{ true DE}, \boldsymbol{X}_j \text{ or } \boldsymbol{X}_k \text{ declared DE})} \\ & \times \mathrm{pr}(\boldsymbol{X}_j \text{ declared DE}|\boldsymbol{X}_j \text{ true DE})\mathrm{pr}(\boldsymbol{X}_j \text{ true DE}), \end{aligned} \tag{5.8}$$

where DE stands for direct effect. The proof is given in Appendix **??**. We make three assumptions in the sequel:

1. The set $\mathcal{J}$ covers all reasonable predictors, in that $p_{j|k}$ is negligible for any $j \notin \mathcal{J}$. We discuss the choice of $\mathcal{J}$ at the end of this section.

2. Each predictor is *a-priori* equally likely to be responsible for a direct effect on $\boldsymbol{Y}$.

3. For each $j \in \mathcal{J}$, pr($\boldsymbol{X}_j$ declared DE$|\boldsymbol{X}_j$ true DE) is the same. In other words the sensitivity of the method does not depend on which predictor happens to possess the effect.

These assumptions allow us to calculate $p_{j|k}$ for each $j \in \mathcal{J}$ as

$$p_{j|k} \propto \frac{\mathrm{pr}(\boldsymbol{X}_k \text{ declared DE}|\boldsymbol{X}_j \text{ true DE, } \boldsymbol{X}_j \text{ or } \boldsymbol{X}_k \text{ declared DE})}{1-\mathrm{pr}(\boldsymbol{X}_k \text{ declared DE}|\boldsymbol{X}_j \text{ true DE, } \boldsymbol{X}_j \text{ or } \boldsymbol{X}_k \text{ declared DE})}, \tag{5.9}$$

then normalising these probabilities to sum to one over the set $\mathcal{J}$.

We now outline the procedure for calculating the right hand side of Equation (**??**). Suppose an effect has been observed in stage I between $\boldsymbol{X}_k$ and $\boldsymbol{Y}$. Let β_k be the size of the direct effect, measured as the change in the estimated effect size if $\boldsymbol{X}_k$ were changed from $\{\boldsymbol{X}_k = 0\}$ to $\{\boldsymbol{X}_k = 1\}$, but all other variables $\boldsymbol{X}_{-k}$ were held constant, and let α_k be the baseline effect size under $\{\boldsymbol{X}_k = 0\}$, with the other variables unchanged, so that

$$\begin{aligned}
\beta_k &= \mathrm{pr}(\boldsymbol{Y}_{\{\boldsymbol{X}_k=1\}} = 1) - \mathrm{pr}(\boldsymbol{Y}_{\{\boldsymbol{X}_k=0\}} = 1), \\
\alpha_k &= \mathrm{pr}(\boldsymbol{Y}_{\{\boldsymbol{X}_k=0\}} = 1).
\end{aligned} \tag{5.10}$$

We estimate α_k and β_k using the association measure z_k, with the indirect effects removed,

$$\begin{aligned}
\hat{\alpha}_k &= \frac{t_{0k} - \mu_{0k} - \sigma_{0k}(z_k - \sum_{k\neq j} \hat{\beta}_k n^{-1/2} \hat{\rho}_{j,k})}{t_{0k}}, \\
\hat{\beta}_k &= \frac{r - t_{0k} + \mu_{0k} + \sigma_{0k}(z_k - \sum_{k\neq j} \hat{\beta}_k n^{-1/2} \hat{\rho}_{j,k})}{t_{1k}} - \hat{\alpha}_k,
\end{aligned} \tag{5.11}$$

see the Appendix **??** for further details.

Suppose that $\boldsymbol{X}_j$ has a true direct effect on $\boldsymbol{Y}$, but this effect has been detected, in stage I, on predictor $\boldsymbol{X}_k$. The effective number of observations that we can use to distinguish between $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$ as the origin of the effect is given by

$$N_E(j,k) = n(\gamma_{0,1} + \gamma_{1,0}),$$

where

$$\gamma_{\omega_1,\omega_2} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(x_{ik}^u = \omega_1, x_{ij}^u = \omega_2), \tag{5.12}$$

i.e. when the two predictors take different values. Evidence towards $\boldsymbol{X}_j$ rather than $\boldsymbol{X}_k$ truly possessing direct effect, the 'truth' in this case, occurs when $(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}) = (0,1,0)$ or $(1,0,1)$ which we suppose happens $ET(j,k)$ times. Evidence towards predictor k rather than predictor j having a direct effect, the incorrect conclusion, occurs when $(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}) = (0,1,1)$ or $(1,0,0)$, which we suppose happens $EF(j,k)$ times. It is clearly possible to observe $EF(j,k) > ET(j,k)$, and is particularly likely for small β_j, small n or large correlation between $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$, resulting in the aforementioned scenario, that $\boldsymbol{X}_k$ is wrongly detected as possessing the direct effect.

Using straightforward algebra (Appendix **??**) we can find,

$$\begin{aligned}
\pi_{EF(j,k)} &= \text{pr}\left[(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}) = (0,1,1) \text{ or } (1,0,0) \mid \boldsymbol{X}_j \neq \boldsymbol{X}_k\right] \\
&= \frac{\gamma_{1,0}\alpha_k + \gamma_{0,1}(1-\alpha_k-\beta_k)}{\gamma_{1,0}+\gamma_{0,1}},
\end{aligned} \tag{5.13}$$

with $\gamma_{\omega_1,\omega_2}$ as in Equation (**??**). For intuition, note that if we assume $t_{0j} = t_{0k}$ this reduces to

$$\hat{\pi}_{EF(j,k)} = \frac{1-\beta_k}{2}.$$

It follows that

$$EF(j,k) \sim \text{Binomial}(N_E(j,k), \pi_{EF(j,k)}), \tag{5.14}$$

so we can use this to calculate

$$\text{pr}\{EF(j,k) > ET(j,k)\} = \text{pr}(EF(j,k) > N_E(j,k)/2)$$

for each $j \in \mathcal{J}$. Noting now the equality of events

$$\{EF(j,k) > ET(j,k)\} = \{\boldsymbol{X}_k \text{ declared DE} \mid \boldsymbol{X}_j \text{ true DE}, \boldsymbol{X}_j \text{ or } \boldsymbol{X}_k \text{ declared DE}\} \quad (5.15)$$

that is a straightforward consequence of the behaviour of the lasso, this allows us to calculate $p_{j|k}$ for $j \in \mathcal{J}$ using Equation (**??**).

There are various ways that $\mathcal{J}$ could be chosen. A cut-off value of ρ could be found so that $\text{pr}\{EF(j,k) > ET(j,k)\}$ is small for $\hat{\rho}_{jk} < \rho$, where ρ_{jk} is the correlation between $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$, i.e. $\boldsymbol{X}_j$ is very unlikely to be the true predictor associated with $\boldsymbol{X}_k$. Alternatively, one could fix the size of $\mathcal{J}$ to, say, the ten predictors that are most highly correlated with $\boldsymbol{X}_k$; or in the spirit of **?**, one could consider using clustering algorithms to select $\mathcal{J}$. In the subsequent work, we adopt the first approach, and choose ρ such that $\text{pr}\{EF(j,k) \geq ET(j,k) \mid \hat{\rho}_{jk} < \rho\} \leq 0{\cdot}01$. Practically, provided conservative bounds are selected when choosing $\mathcal{J}$ the choice of the set is not important. Indeed, one could simply allow $\mathcal{J}$ to contain all the predictors, in this case those predictors that are not highly correlated with $\boldsymbol{X}_k$ would turn out to have a negligible probability of containing the true direct effect.

5.3 Simulations

5.3.1 Introduction

We will now evaluate direct effect testing on the 'ge03d2' dataset taken from the 'GenABEL' package (**?**) in R (**?**).

We study DET by simulating binary responses on the data, with various relationships to the binary predictors. The response for each individual observation, Y_i, is generated according to a Bernoulli distribution with $\text{pr}[Y_i = 1] = \mu_i$, where

$$\mu_i = \alpha + \beta_{j_1} x_{ij_1} + \ldots + \beta_{j_K} x_{ij_K},$$

so that the response is related to a subset of K predictors with indices $\{j_1, \ldots, j_K\}$. The β_{j_k}'s are chosen to represent different effect sizes. To ensure that μ_i is always between zero and one, $|\sum \beta_{j_k}| < 1$. The intercept α is set to

$$\alpha = \frac{1}{2}\left(1 - \sum_{k=1}^{K} \beta_{j_k}\right)$$

to provide approximately equal numbers of observations with $Y_i = 0$ and $Y_i = 1$, and the influential predictor(s) are chosen at random on each simulation. An alternative approach is to generate according to a Bernoulli distribution with $\Pr[Y_i = 1] = \mu_i$, where

$$\log\left(\frac{\mu_i}{1-\mu_i}\right) = \alpha + \beta_{j_1} X_{ij_1} + \ldots + \beta_{j_K} X_{ij_K}.$$

This would avoid the need for a constraint on the βs, but otherwise changes little in terms of the simulations.

Throughout this section we select the significance level for stage I of DET via a Bonferroni correction to achieve a family-wise error rate of 0.05. We record two versions of the DET method — stage I only, in which a find means that the predictor identified by stage I of the method is the true one; and the full DET, where a find means that a true predictor is contained in the set $\mathcal{J}$ associated with a significant effect, and has a probability of being a direct effect at least $0.1p_{\max}$, where $p_{\max}$ is the probability of the most likely predictor in the set. This corresponds to a Bayes factor argument, in that a ratio of greater than 10:1 in favour of the more likely predictor implies strong evidence that the more likely predictor should be preferred. One could change this definition be more or less stringent on which members of $\mathcal{J}$ are allowed to be included: one could choose a different Bayes factor cut-off, or use an absolute probability such as only considering predictors with probabilities of at least 0.05. A 'false find' (ffind) occurs when a significant direct effect is found but there are no associated finds. To be clear, any significant effect identified must be either a find or a false find. When there is more than one true predictor (i.e. $j_K > 1$), there can be more than one find per simulation. A 'complete find' (cfind) is then recorded when all true predictors have been found.

We compare DET with a standard logistic regression with a lasso penalty, where the strength of the penalty is chosen via BIC. We define a 'find' under the standard lasso occurring when a true causal predictor is assigned a non-zero coefficient. Again, when $j_K > 1$ there may be more than one of these per simulation, and a 'complete find' is when all true predictors have been found. Significance testing is not appropriate because the relatively small sample size coupled with the multicollinearity of the dataset means that we do not find coefficients that are significantly different from zero. A lasso 'false find' (ffind) occurs when a non-zero coefficient is assigned to a non-causal predictor. We additionally compare with the 'screen and clean' (S & C) method of Wasserman and Roeder **?**, where the strength of the penalty in the 'screen' stage is also chosen via BIC. In **?** cross validation is used to determine the penalty for the 'screen' stage — this leads to more variables being carried forward to the 'clean' stage as compared with BIC, and hence more true and false finds. Due to the high multicollinearity in this particular dataset, the increase in false finds was particularly damaging for both the lasso and the 'screen and clean' methods, so using BIC seemed to give more favourable results for these methods. The significance level for the 'screen' stage of the S & C procedure is again chosen via a Bonferroni correction to achieve a family-wise error rate of 0.05.

It must be noted that, for the lasso and screen and clean methods, a find is usually declared to have occurred when a non-zero coefficient is found on a predictor highly correlated with the correct predictor. We are, however, considering the case when it is of interest to recover the causal predictor exactly which is frequently of interest in genetic studies. The 'glmpath' function in the 'glmpath' (**?**) library in R (**?**) was used to calculate all lasso paths; all other code was written by the authors and can be obtained upon request.

5.3.2 Simulations on 'ge03d2' Data

The 'ge03d2' dataset contains $n = 897$ subjects, with $p = 7480$ SNPs measured on each subject. We restrict our attention to dominant effects of the SNPs so that, in the usual coding of 0, 1 or 2, we translate all the 2s to 1s. We select two disjoint subsets of the data (subsetting on SNPs not observations), one with $p = 2000$ to study the $p > n$ case, and the other with $p = 400$ to look at the $p < n$ case.

For each of the $p > n$ and $p < n$ cases, we carry out 100 independent simulations on the 'ge03d2' data, where in each case, a subset of predictor(s) is randomly selected, and a response is simulated via various relationships to these predictor(s), as detailed above. Results are then presented as a summary measure over the simulations, so the number of finds recorded is the sum of finds over the simulations. We study here cases of one and four causal predictors, with effect sizes of 10% and 20%. Table **??** gives the results for the $p > n$ case and Table **??** gives the results for the $p < n$ case. The number of finds made by lasso and DET are very similar, despite DET implementing a stringent significance test and lasso merely reporting non-zero coefficients. In addition, the lasso makes a larger number of false finds in general. The screen and clean method achieves similar false find control to DET, but this is at the expense of a far smaller number of true finds. In the case of four predictors, all methods struggle to recover all four predictors on any simulation which corresponds to the number of complete finds being low, but again DET performs better than lasso in terms of recovering the most finds while having similar low false finds as S & C.

Table 5.2: Comparison of lasso and DET finds for $p > n$ case for various effect sizes, for one and four true predictors

Effect size		0.2	0.1	(0.2,0.2,0.2,0.2)	(0.1,0.1,0.1,0.1)
DET (SI* only)	finds	37	2	157	10
	cfinds			1	0
	ffinds	18	5	71	9
DET	finds	44	3	186	11
	cfinds			2	0
	ffinds	11	4	42	8
Lasso	finds	36	1	173	11
	cfinds			1	0
	ffinds	24	5	95	12
S&C	finds	17	0	70	0
	cfinds			0	0
	ffinds	10	7	17	3

*Stage I

5.4 Example

We now illustrate the method on a real dataset. Note that the example has been chosen deliberately to be relatively small and to include only a few predictors to illustrate the results of DET more clearly. In practice, there is no limitation in terms of number of predictors and responses and in fact best results are to be expected in situations with highly correlated

Table 5.3: Comparison of lasso and DET finds for $p < n$ case for various effect sizes, for one and four true predictors

Effect size		0.2	0.1	(0.2,0.2,0.2,0.2)	(0.1,0.1,0.1,0.1)
	finds	49	3	143	11
DET (SI* only)	cfinds			0	0
	ffinds	17	4	61	14
	finds	56	3	176	18
DET	cfinds			3	0
	ffinds	10	4	28	7
	finds	53	6	189	16
Lasso	cfinds			4	0
	ffinds	27	3	89	15
	finds	25	2	84	6
S&C	cfinds			0	0
	ffinds	9	3	24	4

*Stage I

predictors as shown in the previous sections.

The Coronary Risk-Factor Study (?) was carried out in three rural areas in South Africa, in the White Cape region, where incidence of heart disease is particularly high. A subset of the study is analysed extensively in (?). In this subset the binary response, whether or not the subject has heart disease, is measured and 160 cases and 302 controls are collected. Each subject has nine measurements taken as predictors. These are 'sbp' (systolic blood pressure); 'tobacco' (cumulative tobacco); 'ldl' (low density lipoprotein cholesterol); 'adiposity'; 'famhist' (family history of heart disease); 'typea' (type-A behaviour); 'obesity'; 'alcohol' (current alcohol consumption); and 'age' (age at onset, or age of testing for controls). To illustrate DET, we have dichotomized the predictors where necessary, by setting a single threshold level, at an appropriate point where possible: for example, the 'obesity' predictor is originally measured as the Body Mass Index (BMI) and so we have used 30 as the cut-off point, since persons with a BMI exceeding 30 are classed as obese.

We then carry out five analyses on the dichotomized data: the standard single predictor association test, a standard logistic regression, a logistic regression with lasso penalty, the screen and clean method and the direct effect testing method. Results of the single predictor test, the logistic regression and the screen and clean method are given in Table ??. For the screen and clean method, some variables are 'dropped' at the screen stage, so they do not have associated p-values. For the lasso method, four non-zero coefficients were identified — on 'tobacco', 'ldl', 'famhist' and 'age'. For the direct effect testing method, four direct effects were found at the Bonferroni significance level of 0.0056, and the details are in Table **??**. Note that the probabilities do not always sum to one, due to rounding and exclusion of predictors with low ($< 0{\cdot}01$) probabilities, using the cut-off rule specified in Section **??**.

Table 5.4: Comparing p-values calculated via the standard single predictor test and a logistic regression, for the heart disease data

Covariate	Single Predictor	Logistic Regression	S&C
age	$1{\cdot}1 \times 10^{-11}$	$9{\cdot}0 \times 10^{-4}$	$3{\cdot}7 \times 10^{-3}$
famhist	$4{\cdot}8 \times 10^{-9}$	$1{\cdot}1 \times 10^{-5}$	$7{\cdot}0 \times 10^{-3}$
ldl	$4{\cdot}4 \times 10^{-7}$	$6{\cdot}9 \times 10^{-2}$	$3{\cdot}4 \times 10^{-2}$
adiposity	$4{\cdot}4 \times 10^{-6}$	$2{\cdot}4 \times 10^{-1}$	dropped
tobacco	$3{\cdot}2 \times 10^{-7}$	$1{\cdot}0 \times 10^{-1}$	$8{\cdot}3 \times 10^{-2}$
typea	$2{\cdot}3 \times 10^{-1}$	$4{\cdot}3 \times 10^{-2}$	dropped
sbp	$8{\cdot}1 \times 10^{-4}$	$2{\cdot}8 \times 10^{-1}$	dropped
alcohol	$1{\cdot}3 \times 10^{-1}$	$7{\cdot}0 \times 10^{-1}$	dropped
obesity	$1{\cdot}3 \times 10^{-1}$	$3{\cdot}3 \times 10^{-1}$	dropped

Table 5.5: Details from direct effect testing method for heart disease data

Direct Effect p-value	Location	Probability
$4{\cdot}5 \times 10^{-8}$	age	1
$3{\cdot}0 \times 10^{-6}$	famhist	1
$2{\cdot}1 \times 10^{-4}$	tobacco	1
$1{\cdot}3 \times 10^{-3}$	tobacco	$0{\cdot}64$
	ldl	$0{\cdot}31$
	age	$0{\cdot}02$
	typea	$0{\cdot}02$
	adiposity	$0{\cdot}01$

To summarize the findings of the DET analysis, we are virtually certain that 'age', 'famhist' and 'tobacco' have a direct effect on heart disease, which is also reflected in the small p-values in both the logistic regression and the single predictor analysis. There is a possible fourth direct effect, and 'tobacco' re-appears as a possible predictor to possess this direct effect. We interpret this as either evidence of an interaction effect, a direct effect occurring on an unmeasured predictor, or evidence that this fourth direct effect is in fact a false positive. Given that this is the weakest potential effect detected the false positive argument is to be preferred, but it is recommended that expert judgement be used to interpret weaker, ambiguous effects.

5.5 Discussion

In this paper we have introduced a method for binary predictors and response that separates the testing for the presence of a direct effect and the selection of the predictor that produces the effect. This allows, in the first stage, direct effect hypothesis tests to be carried out in the presence of highly correlated predictors without suffering multicollinearity issues. The uncertainty in the assignment of a direct effect to a predictor, caused by the multicollinearity, is taken into account in the second stage, so that the method gives a set of predictors that could represent each direct effect, with probabilities on each predictor in the set. We demonstrate that the method works effectively to find single and multiple direct effects, and compares very favourably with the lasso. Whilst similar methods are available (**?**), DET is unique in offering a probabilistic assessment of which predictors could be associated with the detected effect.

The method easily handles missing data, provided we use the missing completely at random assumption (**?**). Since we deal with cell counts only, a specific observation, x_{ij}, that are missing at any point can be excluded from the count, and therefore no imputation is required. The column totals in Table **??** would then depend on j so we would replace s by s_j, and so forth.

The second stage of the method can be viewed from a Bayesian perspective, by relaxing assumption 2 given in Section **??**, and instead placing a discrete prior on $\mathrm{pr}(\boldsymbol{X}_j \text{ true DE})$. The enforcement of assumption 2 corresponds to a uniform prior.

One of the shortcomings of the method is that it does not allow for multiple levels of the predictor variables. One way to address this issue is by introducing multiple binary predictors for a single discrete predictor.

The model does not currently account for interactions between predictors. Logic regression (**?**) and random forests **?** are existing methods that address this issue; as a consequence of the complexity of such modelling, these methods are algorithmic rather than statistically rigorous. Combining the search for direct effects with a search for interaction effects will be challenging, but represents an important area of future research.

? use Bayesian graphical models to search for relationships between SNPs and a binary response. This naturally takes care of dependencies between SNPs by allowing edges to exist between them. The primary interest is then in edges between SNPs and the binary response. Our method is different in that it separates the uncertainty in size and origin of effects. The relationship of DET to graphical model approaches is still to be investigated fully.

Another interesting point for future investigation are the connections of the introduced method to Genomic Control (**??**) and Delta Centralisation (**?**), which are methods used to account for subpopulation structure or other unobserved confounding effects in a dataset, particularly applied in genetic contexts. This is achieved by assuming the better known noncentral χ^2 null distribution in tests of association, with a noncentrality parameter ν that is common to all tests. This begs the question of whether the direct effect testing method can be used in a similar context, and whether additional power is gained by allowing for a different noncentrality parameter for each test.

In the genetics context, the DET method can be used on either phased or unphased haplotypes. At the point of collection, genetic data is almost always unphased, and it is necessary to use statistical methods to phase the haplotypes. A commonly used method is the PHASE algorithm of **?**. Once the data is phased, we can consider the haplotypes separately, immediately leading to binary data. If haplotypes are unphased, we must consider three possible levels for a SNP (no mutation, single mutation and double mutation). For example, consider a three level predictor $\boldsymbol{X}_j$, taking values 0,1 or 2. Then we introduce two binary predictors, $\boldsymbol{X}_{j_1}$ and $\boldsymbol{X}_{j_2}$. Code $\boldsymbol{X}_{j_1} = 1$ if $\boldsymbol{X}_j \geq 1$ and $\boldsymbol{X}_{j_1} = 0$ if $\boldsymbol{X}_j = 0$; and code $\boldsymbol{X}_{j_2} = 1$ if $\boldsymbol{X}_j = 2$ and $\boldsymbol{X}_{j_2} = 0$ if $\boldsymbol{X}_j \leq 1$. A more general extension that makes use of the multivariate hypergeometric distribution will be investigated in the future. We are also looking into the generalisation to the continuous predictors and response case. Further investigation is needed to compare the performance of DET using the unphased and phased approaches.

5.6 Appendix

Derivation of Equation (??)

To establish Equation (**??**), note that

$$\begin{aligned}\hat{\rho}_{y,j} =& \frac{1}{n}\sum_{i=1}^{n} x_{ij}^{u} y_{i}^{u} \\ =& \frac{a_j d_j - b_j c_j}{\sqrt{srt_{0j}t_{1j}}},\end{aligned}$$

so that

$$\begin{aligned}\sqrt{n}\hat{\rho}_{y,j} =& \sqrt{n}\frac{a_j d_j - b_j c_j}{\sqrt{srt_{0j}t_{1j}}}, \\ =& \sqrt{n}\frac{na_j - t_{0j}s}{\sqrt{srt_{0j}t_{1j}}}.\end{aligned} \tag{5.16}$$

Replacing z_j by the formulae for the mean and standard deviation of the hypergeometric distribution yields

$$\begin{aligned}z_j =& \frac{a_j - t_{0j}s/n}{\sqrt{\frac{srt_{0j}t_{1j}}{n^3}}}, \\ =& \sqrt{n}\frac{na_j - t_{0j}s}{\sqrt{srt_{0j}t_{1j}}}.\end{aligned} \tag{5.17}$$

Therefore Equations (**??**) and (**??**) are equal, proving Equation (**??**).

Derivation of Equation (??)

Write $p_{j|k} = \text{pr}(\boldsymbol{X}_j \text{ true}|\boldsymbol{X}_k \text{ dec.})$, abbreviating in the obvious way. Now by Bayes' Theorem,

$$p_{j|k} \propto \text{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true})\text{pr}(\boldsymbol{X}_j \text{ true}).$$

But

$$\begin{aligned}
\mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}) &= \mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}, \boldsymbol{X}_k \text{ or } \boldsymbol{X}_j \text{ dec.})\mathrm{pr}(\boldsymbol{X}_k \text{ or } \boldsymbol{X}_j \text{ dec.}|\boldsymbol{X}_j \text{ true}) \\
&= \mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}, \boldsymbol{X}_k \text{ or } \boldsymbol{X}_j \text{ dec.})\big\{\mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}) \\
&\quad +\mathrm{pr}(\boldsymbol{X}_j \text{ dec.}|\boldsymbol{X}_j \text{ true})\big\},
\end{aligned}$$

and re-arranging gives

$$\mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}) = \frac{\mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}, \boldsymbol{X}_k \text{ or } \boldsymbol{X}_j \text{ dec.})\mathrm{pr}(\boldsymbol{X}_j \text{ dec.}|\boldsymbol{X}_j \text{ true})}{1 - \mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}, \boldsymbol{X}_k \text{ or } \boldsymbol{X}_j\text{dec.})}.$$

So that

$$p_{j|k} \propto \frac{\mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}, \boldsymbol{X}_k \text{ or } \boldsymbol{X}_j \text{ dec.})\mathrm{pr}(\boldsymbol{X}_j \text{ dec.}|\boldsymbol{X}_j \text{ true})\mathrm{pr}(\boldsymbol{X}_j \text{ true})}{1 - \mathrm{pr}(\boldsymbol{X}_k \text{ dec.}|\boldsymbol{X}_j \text{ true}, \boldsymbol{X}_k \text{ or } \boldsymbol{X}_j \text{ dec.})}$$

as required.

Derivation of Equation (**??**)

Referring to Table **??**, if we were interested in the size of the association between $\boldsymbol{X}_k$ and $\boldsymbol{Y}$, we would estimate this as

$$\begin{aligned}
\mathrm{pr}(\boldsymbol{Y} = 1 \mid \boldsymbol{X}_k = 0) &= \frac{b_k}{t_{0k}} \\
&= \frac{t_{0k} - a_k}{t_{0k}} \\
&= \frac{t_{0k} - \mu_{0k} - \sigma_{0k} z_k}{t_{0k}},
\end{aligned}$$

Removing the indirect effect part, $\sum_{j \neq k} \beta_k n^{-1/2} \hat{\rho}_{j,k}$, immediately yields $\hat{\alpha}_k$ in Equation (**??**).

In order to find the expression for $\hat{\beta}_k$, consider

$$\begin{aligned}
\mathrm{pr}(\boldsymbol{Y}=1 \mid \boldsymbol{X}_k=1) &= \frac{d_k}{t_{1k}} \\
&= \frac{r - t_{0k} + a_k}{t_{1k}} \\
&= \frac{r - t_{0k} + \mu_{0k} + \sigma_{0k} z_k}{t_{1k}}.
\end{aligned}$$

Removing the indirect effect part and subtracting $\hat{\alpha}_k$ then yields the desired result.

Derivation of Equation (??)

Recall that we assume a true direct effect between $\boldsymbol{X}_j$ and $\boldsymbol{Y}$. We then find

$$\begin{aligned}
P_{EF(j,k)} &= \mathrm{pr}\left\{(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}) = (0,1,1) \text{ or } (1,0,0) \mid \boldsymbol{X}_j \neq \boldsymbol{X}_k\right\} \\
&= \frac{\mathrm{pr}\left\{(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}) = (0,1,1)\right\} + \mathrm{pr}\left\{(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}) = (1,0,0)\right\}}{\mathrm{pr}\left\{(\boldsymbol{X}_j, \boldsymbol{X}_k) = (0,1)\right\} + \mathrm{pr}\left\{(\boldsymbol{X}_j, \boldsymbol{X}_k) = (1,0)]\right\}} \\
&= \frac{\gamma_{1,0}\alpha_k + \gamma_{0,1}(1 - \alpha_k - \beta_k)}{\gamma_{1,0} + \gamma_{0,1}},
\end{aligned}$$

where the last line is obtained by writing $\mathrm{pr}\left(\boldsymbol{X}_j, \boldsymbol{X}_k, \boldsymbol{Y}\right) = \mathrm{pr}\left(\boldsymbol{Y} \mid \boldsymbol{X}_j, \boldsymbol{X}_k\right) \mathrm{pr}\left(\boldsymbol{X}_j, \boldsymbol{X}_k\right)$, and using Equation (**??**) for the conditional probabilities of $\boldsymbol{Y}$.

Chapter 6

Recovering Direct Effects in Genetics: A Comparison Study

Abstract

In genetics it is often of interest to discover single nucleotide polymorphisms (SNPs) that are directly related to a disease, rather than just being associated with it. Few methods exist, however, addressing this so-called 'true sparsity recovery' issue. In a thorough simulation study comparing specialised methods, it is shown that for moderate or low correlation (linkage disequilibrium) between SNPs, lasso-based methods perform well at true sparsity recovery, despite not being specifically designed for this purpose. For large correlation, however, more specialised methods are needed. Direct effect testing performs well in all situations, including when the correlation is large.

Keywords: Direct effects, Fine mapping, Lasso, Screen and clean, Stability selection, True sparsity selection, Large p.

6.1 Introduction

In genetic association studies or fine mapping studies where the density of measured single nucleotide polymorphisms (SNPs) in the genome is high, it is of interest to recover SNPs that are directly affecting a response. A SNP with a direct effect is defined as one that has an effect on the response that is not caused by its correlation with any other measured SNP. Those SNPs that are associated with a response but do not have a direct effect are said to have indirect effects. There can be many SNPs with indirect effects, due to the high linkage disequilibrium (LD) or correlation between SNPs on the genome. This LD causes SNPs in proximity to a SNP with a direct effect to be associated with the response. Define the 'true sparsity pattern' to be the set of SNPs (or more generally, predictors) that have a direct effect on the response. The purpose of this paper is to compare the ability of various methods to recover the true sparsity pattern. A contrasting idea is presented by **?**, who use a Bayesian approach to attain models with optimal predictive capabilities.

Suppose a study is carried out with n participants, each typed at p SNPs. Usually the number of SNPs is much larger than the sample size, $p \gg n$. The SNPs and the response are assumed to be related through a generalised linear model,

$$g(E[\boldsymbol{Y}]) = \boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\delta} \tag{6.1}$$

where $\boldsymbol{\eta}$ is an $n \times 1$ response vector explained by an $n \times p$ matrix $\boldsymbol{X}$ (the SNPs) through an unknown $p \times 1$ coefficient vector $\boldsymbol{\beta}$ with $n \times 1$ noise vector $\boldsymbol{\delta}$. On the left hand side $\boldsymbol{Y}$ is the response of interest and $g(\cdot)$ is the link function. An example of a link function is the logit link, $g(x) = \log\left(\frac{x}{1-x}\right)$, which is used when $\boldsymbol{Y}$ is binary. See, for example, **?** for a comprehensive introduction to generalised linear models. Usually only a few of the SNPs under consideration have a direct effect on the response, suggesting $\beta_j = 0$ for most of the predictors. A sparse solution is therefore desired, where only some $\beta_j \neq 0$. The SNPs with direct effects are difficult to find when the predictors are correlated or $p > n$ or both. In these situations there may exist, for example, $\boldsymbol{\beta}'$ and $\boldsymbol{\beta}''$ such that $\boldsymbol{X}\boldsymbol{\beta}' = \boldsymbol{X}\boldsymbol{\beta}''$ but $\boldsymbol{\beta}' \neq \boldsymbol{\beta}''$, causing an identifiability problem (see, for example, **?**). As a consequence, it can be difficult or impossible to identify which of a highly correlated group of predictors possesses an effect; under different permutations of the data, the chosen model will contain different predictors.

The lasso (**?**) is a popular method to identify a sparse set of predictors. It does so by simultaneously shrinking some of the coefficients to zero and estimating effect sizes on the remaining coefficients. It is, however, designed with the goal of prediction in mind rather than true sparsity pattern recovery. If there exists a true direct effect on one of a group of highly correlated predictors (for example, a region on the genome densely populated with SNPs in high LD), the lasso does not attempt to distinguish within the group. Instead, it simply selects one representative member of the group for inclusion in the regression model (**?**). When the objective is prediction this is satisfactory, since having more than one predictor from the group in the model adds little to the predictive capability of the model. For other objectives, such as recovery of the true sparsity pattern, this is not satisfactory. The difficulty in distinguishing between highly correlated predictors also leads to inflated standard errors, which can lead to no predictors being deemed significantly different from zero.

A solution to many of the drawbacks of the lasso is a procedure by **?**, which essentially isolates the significance of an effect from the significance of a predictor. It does this via a two-stage procedure called screen and clean. First, in the 'screen' stage, lasso regression is fitted to one half of the data, so that usually one predictor is selected for each effect. Second, in the 'clean' stage, a standard linear regression is fitted to the other half of the data, using only those predictors selected in the first stage. Therefore, the multicollinearity problem is avoided, and significant *effects* can be identified. A further advantage is that the coefficients estimated in the second stage are no longer shrunk towards zero, so they are not underestimated. The method does not tell us, however, which *predictor*, amongst a correlated group, is responsible for a given effect.

Another method that is considered here is stability selection (**?**). Stability selection works by carrying out a lasso regression multiple times, on bootstrapped samples of the data. These samples are then used to select variables, by including those variables that appear in at least a certain proportion, π_{thr}, of the regressions on the bootstrapped data. This has been shown to be very effective, but it will break down in the presence of highly correlated predictors.

Direct effect testing (**?**) determines the significance of an effect, and retains information on the predictors that could be directly responsible for each effect. In the first stage, one predictor is selected to be responsible for each effect, which allows information about the size and significance of the effect to be obtained. This is similar to screen and clean except that all the data is used at once. In the second stage, a set of predictors that could be the origin of each significant effect are identified, and probabilities are assigned to each of these predictors of being the true direct effect.

We consider here the case where the predictors and the response are binary, because the second stage of DET is currently restricted to such situations. SNPs can be expressed in a binary format despite having three levels, by coding the dominant effect and the recessive effect in two separate predictors. In genetic studies, frequently the responses are also binary (for example, case control studies).

In this paper the performance of lasso, screen and clean, stability selection and direct effect testing in recovering true sparsity patterns is compared, in the binary framework. The comparison is done by simulation. In Section **??** a more detailed summary of the methods under consideration is provided, Section **??** explains how the various simulations are carried out and presents the results. We conclude with a thorough discussion in Section **??**.

6.2 Sparsity Methods

6.2.1 Lasso

The lasso (least absolute shrinkage and selection operator) is introduced first, as all other methods considered are either extended or modified versions of it. **?** introduces the lasso as a regression technique that simultaneously performs variable selection and shrinkage towards zero. It works by minimising a penalised residual sum of squares function,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\ (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{6.2}$$

where λ is a tuning parameter used to control the strength of the penalty. If $\lambda = 0$, the lasso is identical to ordinary least squares; a larger value of λ will cause some coefficients to be zero, which corresponds to selection, and the remaining coefficients to be shrunk towards zero. The tuning parameter λ can be selected by minimising the prediction error through cross validation, or using a model fit criterion such as AIC (**?**) or BIC (**?**).

Since Equation (**??**) includes the residuals, it optimises for prediction rather than true sparsity pattern. As a consequence, consistency is only obtained under some modifications, such as the adaptive lasso (?). In addition, there is an explicit requirement that the correlation between predictors is not too severe. This was first made concrete by ? who call the requirement the irrepresentable condition.

A benefit of the lasso is the efficiency with which it can be calculated. Using the least angle regression algorithm (LARS) of ?, the full lasso path (i.e. the solution of Equation ?? for all values of λ) can be computed in a time comparable to that of a standard regression. On the other hand, a major drawback of the lasso is the difficulty in assigning significance to predictors, particularly in the presence of multicollinearity. Indeed, ? describes this notion as ill-posed; it may be obvious that an effect is present, but the significance is then lost in trying to assign the effect to a specific predictor. The difficulty in finding significant predictors leads to the naive assumption often being made that the sparse model should include all those predictors with nonzero coefficients, and exclude all those with zero coefficients.

6.2.2 Screen and Clean

Screen and clean (?) is an extension to the lasso that uses a two stage approach to overcome two issues with the lasso: the difficulty of significance testing in the presence of a large number of potentially correlated predictors, and the tendency of the lasso to underestimate effect sizes (see, for example, ?). The first stage of screen and clean fits a lasso regression to the first half of the data. This 'screen' stage is intended to produce a collection of potential predictors to carry forward for further testing. The second stage — the 'clean' stage — then fits an ordinary least squares regression to the second half of the data, using only those predictors that had nonzero coefficients in the first stage. This reduced set of coefficients makes significance testing feasible, and avoids effect sizes being underestimated. Significance testing is then carried out with a Bonferroni correction on the reduced set, with considerable success.

In our view, the screen and clean procedure is a method to separate the significance of an effect from the significance of a predictor. We restate the important point, that in a group of highly correlated predictors, one of which being the origin of an effect, the lasso will select one representative member of the group (**?**). In the case of screen and clean, one can therefore see that a variable carried forward to the 'clean' stage may represent an effect, but may not be the predictor with the actual direct effect. It may instead be a predictor that is correlated with the one possessing the direct effect. On the other hand, elimination of all but one predictors in a highly correlated group mitigates the multicollinearity issue. Removal of multicollinearity is one of the major factors leading to the subsequent success of significance testing. Therefore, screen and clean is a method to find significant effects, but not necessarily significant predictors. In the genetics context, one can be optimistic of finding a region of the genotype that is causal to a phenotype, but fine mapping cannot be done.

6.2.3 Stability Selection

Stability selection (**?**) tackles the problem of significance testing for predictors by resampling. The idea is to select predictors that are 'stable', in the sense that their coefficients are nonzero in a certain proportion of lasso regressions carried out on resampled copies of the data. Each sample of the data is generated by selecting half of the observations at random (without replacement). See **?** for discussion of why each sample should contain half of the observations. Lasso regression is then fit to this sample of the data, and the predictors with nonzero coefficients are recorded. This procedure is repeated for each new sample of the data, for a large number, B, of bootstrap samples. Let m_j be the count of times the j^{th} predictor is nonzero. For each predictor $\boldsymbol{X}_j$, $\pi_j = m_j/B$ is then the proportion of times that predictor is present in the fitted model. Predictor $\boldsymbol{X}_j$ is included in the final model if $\pi_j > \pi_{\text{thr}}$, where π_{thr} is a tuning parameter. Stability selection does not provide in itself a method to estimate effect sizes, but one could, for example, fit a linear regression on the variables that it has selected.

A strength of stability selection is that the set of predictors recovered turns out to be insensitive to the choice of π_{thr}, and insensitive to the tuning parameter λ used in each of the lasso regressions (**?**). On the other hand, the nature of stability selection means that it will fail in the presence of highly correlated predictors. Consider a set of 10 similar predictors, one of which has a large effect on the response. Then each predictor may have $\pi_j \approx 1/10$.

This π_j would be smaller than any reasonable threshold π_{thr}, meaning that none of the 10 predictors would be selected, and the model would fail to include the true effect.

6.2.4 Direct Effect Testing

Direct effect testing (DET) (**?**) is a method that detects direct effects, and calculates a probability distribution giving the probabilities that each predictor is the true origin of each direct effect. The first stage of DET identifies direct effects by using lasso regression to separate direct effects from indirect effects. To calculate the significance of an effect, each direct effect is attributed, automatically by the lasso, to a specific predictor. The lasso has the property that a significant effect is usually detected on one single predictor (see, for example **?**). The lasso does not recover the true sparsity pattern *per se* — the predictor to which the effect has been attributed may not be the true origin of the effect.

The second stage of DET then incorporates the uncertainty in the predictor that is the true origin of each direct effect. We give a sketch of stage two of DET — see **?** for more detail. Consider a model with only two predictors, $\boldsymbol{X}_j$ and $\boldsymbol{X}_k$, which are highly correlated with each other, and highly correlated with a response, $\boldsymbol{Y}$. In truth, one of the predictors, $\boldsymbol{X}_j$ or $\boldsymbol{X}_k$, has a direct effect on the response, but it is not known which one. Suppose in the first stage of DET, a significant effect is discovered on predictor $\boldsymbol{X}_k$. Then the probabilities of interest are

$$\Pr[\boldsymbol{X}_k \text{ true effect}|\text{Effect observed in stage one on } \boldsymbol{X}_k]$$

and

$$\Pr[\boldsymbol{X}_j \text{ true effect}|\text{Effect observed in stage one on } \boldsymbol{X}_k]$$

Simple manipulations using Bayes theorem allow both of these probabilities to be obtained when the predictors are binary. The procedure then naturally generalises to any number of binary predictors.

DET therefore provides a novel method to carry out fine mapping by distinguishing between direct and indirect effects, and quantifying the uncertainty associated with this distinction. The main drawback of the method is that stage two can only be applied to binary predictors, which is the reason that the simulations carried out in this paper are restricted to the binary case.

6.3 Design and Results of Simulations

A range of simulations are carried out to study the properties of the methods introduced above. Situations involving strong and weak correlations, including both serially correlated data and clustered data are considered, and consistency properties of the methods on these simulated data are investigated. Additionally, the performance of the methods on some covariate patterns taken from real data with a simulated response are compared. Finally, the performance of the methods is investigated on genetic type data generated using the 'Fregene' software (**??**). We begin by explaining how the various methods are applied, how significance is determined, and introduce some of the measures used to compare the methods.

For the lasso, the penalty is selected using the BIC (**?**), due to increased speed over cross validation. Results based on the BIC were uniformly superior to the AIC in our studies. Since determining significance is difficult for the lasso, a significant find is recorded when a predictor receives a nonzero coefficient.

For screen and clean, the data is randomly divided in half for each simulation. The first half of the data is used to fit the lasso (screen stage), where the penalty is selected by BIC, for the reasons given before. For the clean stage, ordinary least squares is carried out on the nonzero coefficients, and significance is determined using a Bonferroni correction, as suggested in **?**.

For stability selection, 100 samples of the data of size $n/2$ are sampled without replacement on each simulation. Lasso regression is then fit to each sample, but this time AIC is used to determine the lasso penalty, as this dominated the BIC results on the simulations considered. The threshold is set to $\pi_{\text{thr}} = 0.75$, meaning that a predictor must be selected in at least 75 out of the 100 lasso regressions to be declared significant. **?** claim that any value of π_{thr} in the range $(0.6, 0.9)$ gives similar results, justifying this choice.

For direct effect testing, significance of each predictor is calculated using a Bonferroni correction. For stage two of the procedure, let $p_{\max}$ be the probability of the best contender for the origin of a given direct effect, then a 'find' is defined as any other predictor that has a probability of origin that is at least 10% as large as the best contender, i.e. $0.1p_{\max}$. Results from DET using stage one only are also recorded, so that the circumstances under which stage two gives additional benefit can be studied.

It is emphasised that the comparisons are not designed to be an indicator of which methods are universally best. Each method is designed to deal with different scenarios, and here only the ability of the methods to recover true sparsity patterns is considered. Moreover, it is difficult to make the comparisons fair, for example DET (including stage two) can have multiple chances to make a find per significant effect identified, while formal significance testing is not used in declaring significance for the lasso.

Whichever method is used to obtain the data, it is necessary to generate the response artificially through some dependence on one or more of the predictors. The response for each individual observation, Y_i, is generated according to a Bernoulli distribution with $\Pr[Y_i = 1] = \mu_i$, where

$$\mu_i = \alpha + \beta_{j_1} X_{ij_1} + \ldots + \beta_{j_K} X_{ij_K},$$

so that the response is related to a subset of K predictors with indices $\{j_1, \ldots, j_K\}$. The β_{j_k}'s are chosen to represent different effect sizes. So that μ_i is always between zero and one, $|\sum \beta_{j_k}| < 1$. The intercept α is set to

$$\alpha = \frac{1}{2}\left(1 - \sum_{k=1}^{K} \beta_{j_k}\right)$$

to provide approximately equal numbers of observations with $Y_i = 0$ and $Y_i = 1$, and the influential predictor(s) are chosen at random on each simulation. An alternative approach is to generate according to a Bernoulli distribution with $\Pr[Y_i = 1] = \mu_i$, where

$$\log\left(\frac{\mu_i}{1 - \mu_i}\right) = \alpha + \beta_{j_1} X_{ij_1} + \ldots + \beta_{j_K} X_{ij_K}.$$

This would avoid the need for a constraint on the βs, but otherwise changes little in terms of the simulations.

If a method finds a predictor that was truly used to generate the response, this is called

Table 6.1: Summary of find definitions

Name	Description
True find	True predictor found
Strong true find	All true predictors found
False find	False predictor found
Strong false find	At least one false predictor found
Perfect find	Strong true find and no strong false find
FDR	False discovery rate (false finds / total finds)

a 'true find'; consequently when more than one predictor is used to generate the response, there can be more than one true find per run on a simulation. On the other hand any predictors that are declared significant and were not used to generate the response are declared 'false finds'; consequently there can be more than one false find per run on a simulation. For the case of DET there can be multiple finds per significant effect. In this case a significant effect is a true find providing at least one true predictor is a find associated with that effect. A 'strong false find' is recorded whenever at least one incorrect predictor is included in the model. In cases where the response explicitly depends on more than one predictor, a 'strong true find' is recorded when all the causal predictors are recovered. A 'perfect find' is recorded when the model includes all causal predictors and no other predictors (i.e. simultaneous occurrence of a strong true find and nonoccurrence of a strong false find). Finally, the standard definition of false discovery rate (FDR),

$$\text{FDR} = \frac{\text{Number of false finds}}{\text{Total number of finds}},$$

is also used. For convenience, the definitions above are summarised in Table **??**.

The 'glmpath' function in the 'glmpath' (**?**) library in R (**?**) was used to calculate all lasso paths. All other code was written by the authors.

6.3.1 Serially Correlated Data

Data Generation

To generate serially correlated data, denoting the strength of the correlation by ρ, the approach used is:

1. Generate random binary realisations for $X_{i,1}$, $i = 1, \ldots, n$, according to Bernoulli(0.5)

where n, the sample size, is set to $n = 1000$ here.

2. For $j = 2, \ldots, p$, where p is the number of predictors, set to $p = 400$ here, generate binary realisations for $X_{i,j}$, $i = 1, \ldots, n$, via

$$X_{i,j} = \begin{cases} X_{i,j-1} & \text{with prob. } \rho \\ 0 & \text{with prob. } (1-\rho)/2 \\ 1 & \text{with prob. } (1-\rho)/2. \end{cases}$$

3. Randomly assign a causal predictor with effect size 0.2, repeating this ten times on each dataset.

4. Repeat steps 1–3 100 times to obtain 100 different datasets, and 1000 simulations in total.

This procedure is repeated for a range of serial correlations, from $\rho = 0$ to $\rho = 0.99$.

Results

Figure **??** illustrates the simulated true finds, perfect finds, false finds and FDRs for serial correlations ranging from 0 to 1. Most of the methods deteriorate in fairly similar ways as the correlation approaches one. The exception is DET including stage two, whose perfect find rate and FDR do not become worse as the correlation becomes very large. Indeed, once the correlation becomes close to one, DET has the highest number of perfect finds and the lowest FDR of all the methods. This is a consequence of the second stage of DET accounting for the high correlation by giving a large set of potential predictors for a detected effect. It is only once $\rho > 0.75$, however, that including stage two of DET gives additional benefits over stage one alone. For small to moderate correlation, stability selection gives the highest perfect find rate and lowest FDR.

Results are also given for the situation where the response is independent of the predictors, i.e. there are no causal predictors. The false find rates for this situation are given in Figure **??**. These are roughly constant for lasso, screen and clean, and stability selection, with the latter being much lower than the other two. Interestingly, the number of false finds

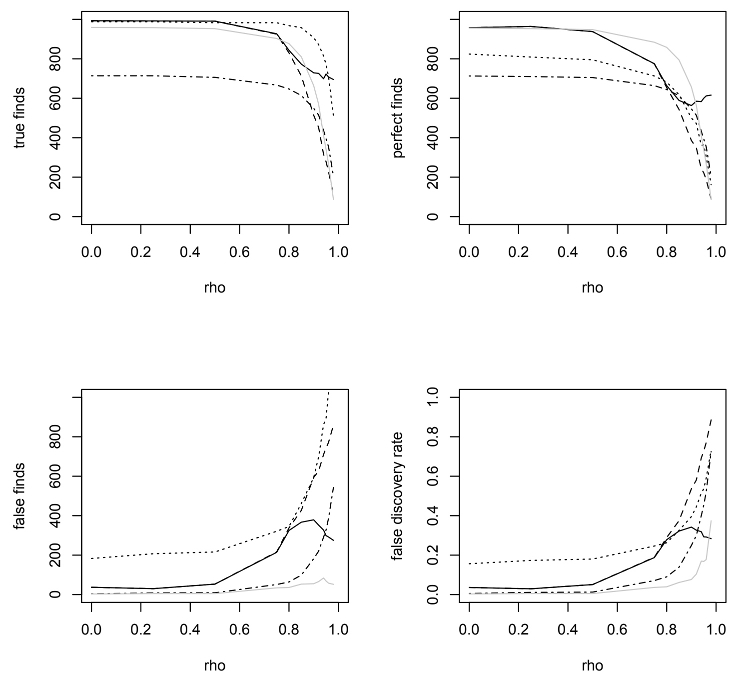

Figure 6.1: Serial correlation: Comparing true finds (top left), perfect finds (top right), false finds (bottom left) and false discovery rate (bottom right) for different values of ρ. Heavy solid line — DET, dashed line — DET (stage one only), dotted line — lasso, dot-dashed line — screen and clean, light solid line — stability selection.

for DET decreases as the serial correlation ρ increases. Note that since there is no true effect, stage two of DET has no effect here.

Consistency

The consistency properties of the different methods in the serial correlation framework are now studied. Unlike Section **??** the correlation is fixed and the sample size, n, varied. Datasets are generated using the same method as Section **??** for the largest sample size under consideration, $n = 10000$. Simulations for smaller sample sizes are done by taking subsets of the $n = 10000$ observations. This ensures that the simulations are as similar as possible.

Figure **??** gives true finds, perfect finds, false finds and FDRs for the methods with high correlation, $\rho = 0.95$. Figure **??** gives the same for correlation $\rho = 0.5$. In both cases the

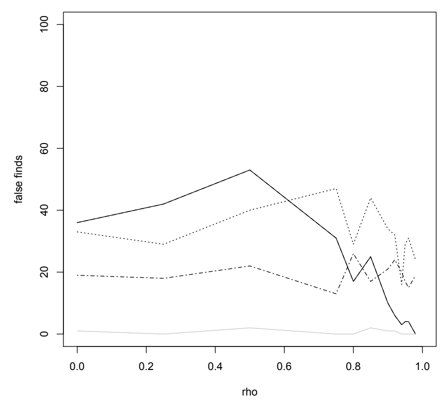

Figure 6.2: Serial correlation: Comparing false find rates for different values of ρ, when there is no true causal effect. Heavy solid line — DET, dotted line — lasso, dot-dashed line — screen and clean, light solid line — stability selection.

number of predictors $p = 400$.

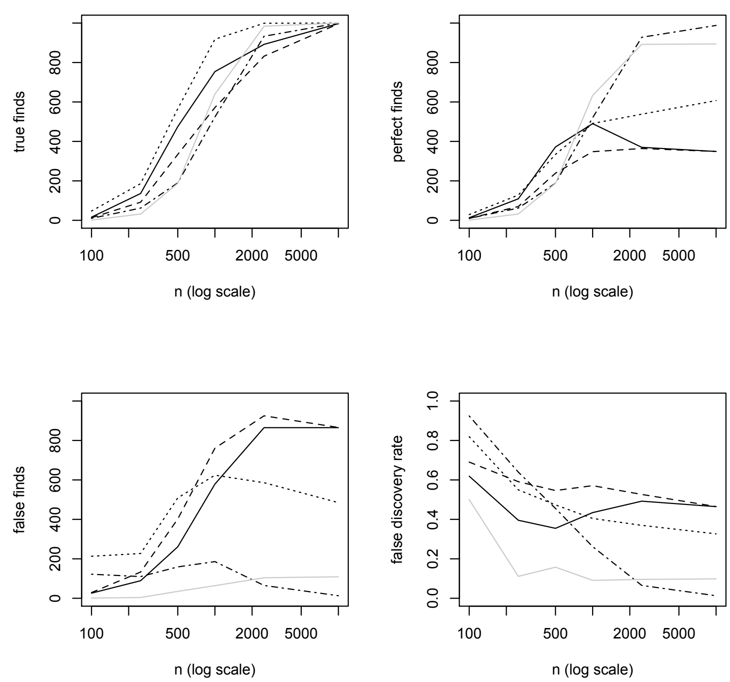

Figure 6.3: Comparing true finds (top left), perfect finds (top right), false finds (bottom left) and false discovery rate (bottom right) for differing sample sizes n, on the log scale. Serial correlation $\rho = 0.9$. Heavy solid line — DET, dashed line — DET (stage one only), dotted line — lasso, dot-dashed line — screen and clean, light solid line — stability selection.

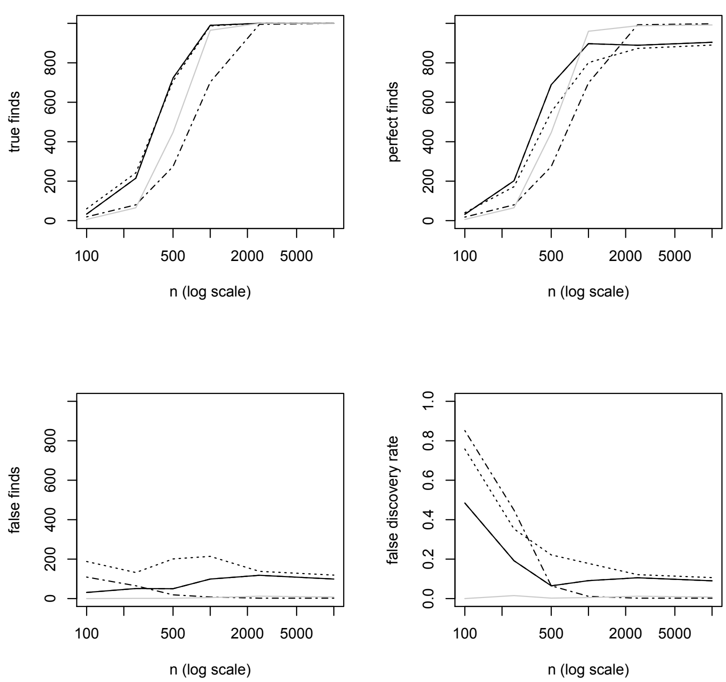

Figure 6.4: Comparing true finds (top left), perfect finds (top right), false finds (bottom left) and false discovery rate (bottom right) for differing sample sizes n, on the log scale. Serial correlation $\rho = 0.5$. Heavy solid line — DET, dashed line — DET (stage one only), dotted line — lasso, dot-dashed line — screen and clean, light solid line — stability selection.

Screen and clean has the best consistency properties in the high correlation scenario, since the perfect find rate increases, and the false discovery rate decreases as the sample size increases. Stability selection also performs well in the high correlation scenario since the perfect find rate increases, and the false discovery rate is controlled (although it does not appear to decrease as fast when n increases). Direct effect testing performs poorly as sample size increases in the large correlation case; a large number of false finds are generated for large n, and the perfect find rate does not increase. Lasso does slightly better, but the increase in the perfect find rate is much slower than for stability selection and screen and clean.

For the low correlation case (Figure **??**), all the methods considered have good consistency properties. DET outperforms the other methods in terms of perfect finds and FDR for small n; stability selection and screen and clean are the better performers on these measures for larger sample sizes n.

6.3.2 Clustered Data

Data Generation

As an alternative to serial correlation, clustered data is considered. The data here is generated with $p = 400$ predictors, divided into clusters of size k, for a range of cluster sizes from $k = 1$ to $k = 10$. Within cluster correlation is set to ρ and there is independence between clusters. This is repeated for different correlations ρ. The data generation procedure is as follows:

1. Generate random binary realisations for $X_{i,j}$, $i = 1, \ldots, n$, where the sample size is set to $n = 1000$, according to Bernoulli(0.5) for the first $\boldsymbol{X}_j$ in each cluster. Hence this step is carried out p/k times.

2. For each subsequent $\boldsymbol{X}_j$ in each cluster, generate binary realisations for $X_{i,j}$, $i = 1, \ldots, n$ via
$$X_{i,j} = \begin{cases} X_{i,j-1} & \text{with prob. } \rho \\ 0 & \text{with prob. } (1-\rho)/2 \\ 1 & \text{with prob. } (1-\rho)/2. \end{cases}$$
   Hence this is carried out $k - 1$ times in each cluster.

Results

Figure **??** visualises the results of various cluster sizes, for a within cluster correlation $\rho = 0.9$. Stability selection does a surprisingly good job of controlling the FDR and maintaining a large number of perfect finds, out-performing all the other methods.

Figure **??** repeats the cluster size study but with the within cluster correlation now set to $\rho = 0.95$. The larger correlation causes stability selection to struggle to make perfect finds, but it does continue to control the FDR. Direct effect testing now achieves the largest number of perfect finds, while having the second best control of FDR.

Consistency of the various methods was also considered in the clustering framework. The results are almost identical as the consistency results in the serial correlation framework, so

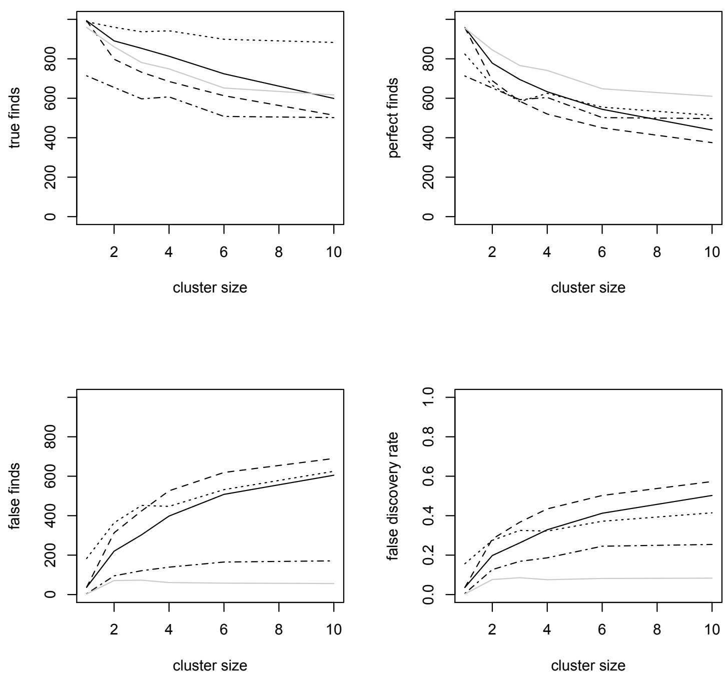

Figure 6.5: Comparing true finds (top left), perfect finds (top right), false finds (bottom left) and false discovery rate (bottom right) for different values of cluster sizes, within cluster correlation 0.9. Heavy solid line — DET, dashed line — DET (stage one only), dotted line — lasso, dot-dashed line — screen and clean, light solid line — stability selection.

the details are omitted.

6.3.3 Real Covariate Patterns

It is important to consider how the methods behave when covariate patterns from a real dataset are used. Here, the 'ge03d2' data taken from the 'GenABEL' package (**?**) in R (**?**) is used. This dataset contains $n = 897$ subjects, with $p = 7480$ SNPs measured on each subject. Attention is restricted to dominant effects of the SNPs so that, in the usual coding of 0, 1 or 2, all the 2's are translated to 1's. Various subsets of the SNPs are considered, including two non-overlapping subsets with $p = 400$ and $p = 2000$, to allow both the $p < n$ and $p > n$ cases to be considered. A binary response is then simulated based on various relationships to the predictors.

Tables **??**, **??** and **??** present the data with real covariate patterns. Three situations

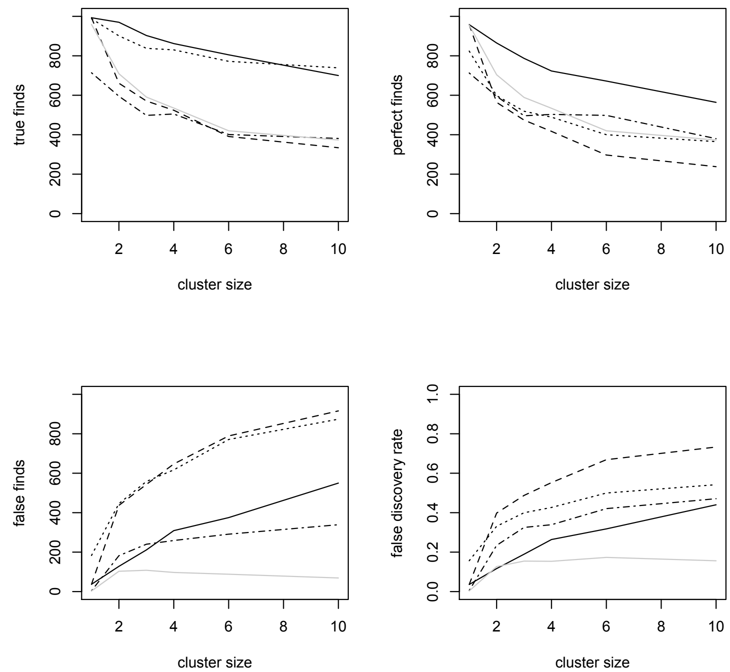

Figure 6.6: Comparing perfect finds (top left), false discovery rate (top right), true finds (bottom left) and false finds (bottom right) for different values of cluster sizes, within cluster correlation 0.95. Heavy solid line — DET, dashed line — DET (stage one only), dotted line — lasso, dot-dashed line — screen and clean, light solid line — stability selection.

are considered — a single predictor having an effect of size 0.2 on a response (Table **??**), two predictors both having an effect of size 0.2 on the response (Table **??**), and a single predictor with a weaker effect, size 0.1, on the response (Table **??**). For the larger effect sizes, DET performs the best in terms of both maximising the number of perfect finds and minimising the FDR. Stage 2 of DET is beneficial regardless of the cluster size. These outcomes are similar to the results for the clustered and serially correlated simulations, for sufficiently large correlation. Tables **??** and **??** show that stability selection achieves similar FDR control to DET but is less powerful. There is cohesion, therefore, between the results based on the simulated data, and the results obtained from the real covariate patterns. For the smaller effect size, stability selection achieves the lowest FDR, but this is at the expense of a low perfect find count. Lasso and DET share the highest perfect find counts in this case (Table **??**).

Table 6.2: Simulations with single effect, size 0.2, real covariate patterns

eff=0.2	True finds	False finds	Strong false finds	Perfect finds	FDR
DET (S1)	457	159	148	426	0.26
DET	545	71	69	505	0.12
Lasso	542	261	216	413	0.33
S&C	231	89	89	231	0.28
Stab. sel.	329	50	49	315	0.13

Table 6.3: Simulations with two effects, both of size 0.2, real covariate patterns

eff=0.2,0.2	True finds	Strong true finds	False finds	Strong false finds	Perfect finds	FDR
DET (S1)	800	153	331	293	487	0.29
DET	991	240	140	132	648	0.12
Lasso	1030	297	540	408	405	0.34
S&C	436	68	144	138	342	0.25
Stab. sel.	657	123	101	100	486	0.13

Similar results are observed for another data set with $p = 2000$. The effect size of 0.2 is chosen as it demonstrates most clearly the differences between the methods, although other smaller effect sizes gave comparable results.

Finally, for comparison, Table **??** gives the number of false finds and strong false finds that are made when the response is independent of the predictors (for the $p = 400$ case). All methods (but particularly stability selection) control the false find rates well in this situation.

6.3.4 Genetic-type Data

Data Generation

We now investigate genetic-type data, generated using the 'Fregene' software (**??**). This software allows forward-in-time simulations of genetic type data, making it a versatile tool for simulations. The particular data used in this illustration was generated based on a single

Table 6.4: Simulations with single effect, size 0.1, real covariate patterns

eff=0.2	True finds	False finds	Strong false finds	Perfect finds	FDR
DET (S1)	33	36	32	31	0.52
DET	43	26	22	41	0.38
Lasso	49	42	39	41	0.46
S&C	11	22	22	11	0.67
Stab. sel.	9	3	3	8	0.25

Table 6.5: Simulations with no effect, i.e. response independent of predictors, real covariate patterns

eff=0	False finds	Strong false finds
DET (S1)	21	21
DET	21	21
Lasso	33	32
S&C	24	24
Stab. sel.	1	1

Table 6.6: Simulations with no effect, i.e. response independent of predictors, real covariate patterns

eff=0	False finds	Strong false finds
DET (S1)	5	4
DET	5	4
Lasso	4	4
Fisher's Exact	8	5

chromosome, of length 3 megabases, with 10000 chromosomes in the initial population. The simulation was allowed to run for 200000 generations, with a mutation rate of 2.3×10^{-8}. Two thousand haplotypes were generated, and carried forward for analysis. There were a total of 12835 SNPs; after discarding any SNPs with minor allele frequency less than 10%, 2545 SNPs remained. These 2545 SNPs were used in the subsequent analysis.

Results

We compared the performance of DET (with and without stage 2), lasso and Fisher's exact test on this dataset. Here, the methods were empirically calibrated so that, in the case when a response is generated independent of the predictors, the strong false find rate is controlled at approximately 5%. To illustrate the control achieved, Table **??** gives the numbers of strong false finds and false finds made in 100 simulations, where the response is generated independent of the predictors.

We also carried out 100 simulations where a binary response is generated, with a single predictor having an effect of size 0.2. Table **??** gives the results for this case.

False finds are, of course, no longer at the 5% level because SNPs correlated with the causal SNP are likely to be declared significant. Unsurprisingly, the performance of Fisher's exact test is poor, with a large number of false finds. Lasso, and DET with stage one

Table 6.7: Simulations with single effect, size 0.1, real covariate patterns

eff=0.2	True finds	False finds	Strong false finds	Perfect finds	FDR
DET (S1)	35	161	82	10	0.82
DET	82	114	50	42	0.58
Lasso	33	57	45	17	0.63
Fisher's exact	100	7831	99	1	0.99

only, are comparable in terms of performance. DET, including stage 2, has the lowest false discovery rate and a perfect find rate more than double that of lasso. The performance of the lasso is greatly improved by controlling its false finds in this way. Note the similarity of these results with those in Section **??**.

6.4 Discussion

This paper presents a thorough simulation study of the performance of various methods at recovering the true sparsity pattern, specifically with application in genetic studies. Despite many of these methods not being designed for the objective of true sparsity recovery, performance of all the methods is good, provided the correlation in the predictors is not too high. Once the correlation becomes high, specialised methods such as direct effect testing are needed. Direct effect testing also out-performs the other methods considered in scenarios where the sample size is small. Direct effect testing, however, does not have good consistency properties; in our simulations screen and clean appears to have the best consistency properties.

We were surprised in particular with how well stability selection performed. At first glance, it seems that because of the bootstrapping approach used, even moderate correlation may cause problems for the method. Whilst it does begin to fail once the correlation is very large, datasets where neighbouring correlations are as high as $\rho = 0.95$ are handled well. Indeed, until the correlation reaches this high level, stability selection also controls the type one error. This was proved for the case of exchangeable coefficients in Theorem 1 of **?**, the work here represents further empirical evidence that similar control is achieved in many non-exchangeable situations.

As the density of information collected increases, measured predictors will become increasingly correlated. So methods that are specifically designed to handle this, such as direct effect testing, will become more important. Direct effect testing, however, only deals with binary predictors and response. Therefore, novel methods that can handle highly correlated continuous variables are needed.

Chapter 7

More on Direct Effect Testing

7.1 Introduction

This Chapter considers various extensions and alternatives to direct effect testing (DET). The general theme of the Chapter is to make suggestions on how DET can be generalised to the continuous predictors and response case.

In Section **??** we illustrate that the unpenalised version of the correlated residuals model has the same ordinary least squares solution as a normal linear regression. Once a penalty is applied, however, we illustrate that the correlated residuals version of the lasso and the standard version of the lasso do not lead to the same solution. In Section **??** we relate the lasso used in DET to the standard lasso (**?**) and the Dantzig selector (**?**). We then suggest how these relationships could be used to derive asymptotic results for DET. In Section **??** we focus on stage 2 of DET, and derive a fully Bayesian alternative based on the work of **?**. Such a procedure generalises naturally to the continuous case. We compare DET stage 2 with the Bayesian approach through simulation.

Throughout this Chapter, we use the same notation as **?** (Chapter **??**).

7.2 Relationships between Correlation Model and Ordinary Least Squares

Consider the standard linear regression

$$Y_i = \sum_{j=1}^{p} \beta_j X_{ij} + \delta_i,$$

where for observations $i = 1, \ldots, n$, Y_i and X_{ij} have exactly the same meaning as in the DET framework, so that Y_i represents the response and X_{ij} is the value of the j^{th} predictor variable. Additionally here, δ_i is the residual and β_j is the coefficient for the j^{th} predictor. Recall from Chapter 5 the DET model based on correlation structure:

$$\hat{\rho}_{y,j} = \sum_{k=1}^{p} \hat{\rho}_{j,k}\theta_k + \epsilon_j, \tag{7.1}$$

First, we note that Equation (**??**) is equivalent to this standard linear regression. This is not a new observation. It appears, for example, in **?** who describes Equation (**??**) as 'the normal equations of the method of least squares in a slightly disguised form'. The author's proof given here also illustrates the role of the error term, ϵ_j.

Proposition 1. *The model*

$$\hat{\rho}_{y,j} = \sum_{k=1}^{p} \hat{\rho}_{j,k}\theta_k + \epsilon_j, \tag{7.2}$$

is equivalent to the model

$$Y_i = \sum_{j=1}^{p} \beta_j X_{ij} + \delta_i. \tag{7.3}$$

Moreover, for each $j = 1, \ldots, p$, $\beta_j = \theta_j$, and $\epsilon_j = \hat{\rho}_{\delta,j}$, where $\hat{\rho}_{\delta,j}$ is the observed correlation between the residuals and predictor $\boldsymbol{X}_j$.

Proof. By definition

$$\hat{\rho}_{y,j} = \frac{1}{n} \sum_{i=1}^{n} X_{ij} Y_i,$$

and therefore replacing Y_i using Equation (**??**) gives

$$\begin{aligned}\hat{\rho}_{y,j} &= \frac{1}{n}\sum_{i=1}^{n} X_{ij}\left(\sum_{j=1}^{p}\beta_j X_{ij} + \delta_i\right), \\ &= \sum_{k=1}^{p}\hat{\rho}_{j,k}\beta_k + \frac{1}{n}\sum_{i=1}^{n} X_{ij}\delta_i.\end{aligned}$$

and by comparison with Equation (**??**), $\beta_k = \theta_k$ and $\epsilon_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}\delta_i = \hat{\rho}_{\delta,j}$. □

As a consequence of Equation (**??**) corresponding to the normal equations for Equation (**??**), it is clear that the least squares solutions are identical (**?**). On the other hand, the lasso regression solutions for the two equations will not be the same. Call a lasso regression carried out on Equation (**??**) *standard lasso*, and a lasso regression carried out on Equation (**??**) *correlated lasso*. The lasso paths generated using these two equations are different, despite the OLS solutions being the same. We show this through an example. We generate 200 observations, where each observation consists of a response and four predictors, but all responses and predictors are simply generated independently from a standard normal distribution. Figure **??** shows the lasso paths generated for these frameworks.

The value of making these comparisons lies in illustrating the differences and similarities of DET to standard approaches. The fact that the OLS solutions for the correlation model and the normal linear model are the same is reassuring in terms of consistency of the correlation model approach. The open questions are what causes the difference in the lasso paths between the two methods, and when or why should one method be preferred over the other?

7.3 Relationships between Correlation Model Lasso, Standard Lasso and the Dantzig Selector

Lasso (**?**) can be written as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, \boldsymbol{\beta}\|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{7.4}$$

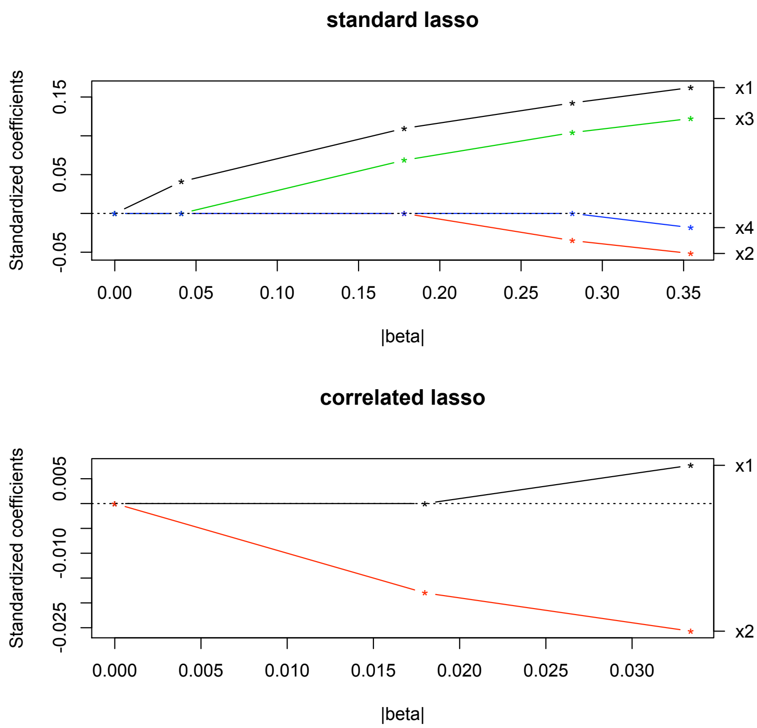

Figure 7.1: Comparison of the lasso paths on simulated independent data (i.e. no true effects between predictors and response) for standard lasso (top) and correlation lasso (bottom)

The Dantzig selector (**?**) can be written as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\ \boldsymbol{\beta}\|\boldsymbol{X}'(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta})\|_\infty + \lambda\|\boldsymbol{\beta}\|_1. \tag{7.5}$$

For the lasso-based method used within DET, Equation (**??**) written in matrix notation gives

$$\boldsymbol{R}^y = \boldsymbol{R}\boldsymbol{\theta} + \boldsymbol{\epsilon},$$

or

$$n^{-1}\boldsymbol{X}'\boldsymbol{Y} = n^{-1}\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}.$$

So by comparison with Equation (**??**) it is easy to see that lasso within DET is equivalent to

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\ \|\boldsymbol{X}'(\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{\beta})\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{7.6}$$

So it appears that the lasso within DET combines features of the standard lasso and of the Dantzig selector. This is interesting in the context of recent work that has focussed on the relationship between the lasso and the Dantzig selector. **?** and **?** establish that under certain sparsity conditions, i.e. when most of the coefficients are zero, the lasso and Dantzig selector yield the same solution. When the solutions are not the same, the Dantzig solution will always be sparser than the lasso. See **?** for some illuminating graphical representations of the comparison between the two methods. This leads to the conjecture that the solution produced by the lasso within DET is 'sandwiched' by the standard lasso and the Dantzig selector, in the sense that it is guaranteed to be at least as sparse as the lasso solution, but not more sparse than the Dantzig solution. Given the extensive theoretical results available for the lasso (see, for example, **??**), and corresponding results for the Dantzig selector (**?**), it could be possible to derive similar results for the correlation model lasso using a sandwich inequality.

7.4 DET stage 2: Comparison with a Bayesian Approach and Evaluation

In this Section we develop a Bayesian alternative to stage 2 of DET, based on ideas from **?**. We then compare the Bayesian approach with DET stage 2 in terms of the consistency of the two methods, and their ability to assign high probabilities to truly causal predictors and low probabilities to predictors with indirect effects.

7.4.1 Bayesian Alternative to DET Stage 2

Following similar ideas to those in **?**, we develop a Bayesian alternative to stage 2 of DET. Assume that there is one effect on a binary response $\boldsymbol{Y}$ in a group of p (binary) predictors, $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p\}$, but we do not know which predictor has the effect. Due to the high correlation between the predictors, many predictors will be associated with the response. If we assume that predictor $\boldsymbol{X}_j$ is the origin of the effect, the likelihood is given by

$$L_j(\mu_0, \mu_1) = \prod_{i:X_{ij}=0} \left\{\mu_0^{Y_i}(1-\mu_0)^{1-Y_i}\right\} \prod_{i:X_{ij}=1} \left\{\mu_1^{Y_i}(1-\mu_1)^{1-Y_i}\right\},$$

where $\mu_\omega = \Pr(Y_i = 1|X_{ij} = \omega)$. Assume that μ_0 and μ_1 both have Uniform$[0, 1]$ priors, and the location of the true causal predictor has a uniform prior so that $\Pr[\boldsymbol{X}_j \text{ causal}] = 1/p$ for each predictor. Then the marginal likelihood for predictor $\boldsymbol{X}_j$ can be computed exactly as

$$ML(j) = \frac{a_j!b_j!c_j!d_j!}{(t_{0j}+1)!(t_{1j}+1)!},$$

and the posterior probability of each predictor $\boldsymbol{X}_j$ being causal is

$$PP(j) = \Pr[\boldsymbol{X}_j \text{ causal}|\boldsymbol{Y}] = \frac{ML(j)}{\sum_{k=1}^{p} ML(k)}.$$

The Bayesian approach will generalise more naturally to the continuous predictor and response case than the approach presented as DET stage 2. This could be done by reformulating the likelihood, then using a Gibbs sampler to approximate the marginal likelihood.

7.4.2 Comparison and Verification of Stage 2 Procedures

We now study the quality and consistency of DET stage 2 and the Bayesian alternative on simulated data using the following algorithm:

1. Generate $n = 100$ observations of $p = 10$ binary predictors that are serially correlated with a randomly chosen correlation ρ.

2. Generate a binary response that depends on the fifth predictor with effect size 0.2.

3. Calculate $PP(j)$ for each $j = 1, \ldots, 10$ using both the Bayesian and DET stage 2 approaches.

4. Record $PP(5)$ in a vector $\boldsymbol{S}$ — the 'successes', and $PP(j), j \neq 5$ in a vector $\boldsymbol{F}$ of 'failures'.

5. Repeat steps 1–4 multiple times, T times say, concatenating the $PP(j)$'s in the same vectors $\boldsymbol{S}$ and $\boldsymbol{F}$. Hence $\boldsymbol{S}$ is a vector of length T and $\boldsymbol{F}$ is a vector of length $9T$.

6. Consider the partition $P_1 = [0, 0.05)$, $P_2 = [0.05, 0.10)$, ..., $P_{20} = [0.95, 1.00)$. Let $\boldsymbol{S}_{P_k}$ be the subvector of $\boldsymbol{S}$ whose probabilities are contained within the k^{th} partition, and let $\boldsymbol{F}_{P_k}$ be the subvector of $\boldsymbol{F}$ whose probabilities are contained within the k^{th} partition. Let $|\boldsymbol{X}|$ denote the length of a vector $\boldsymbol{X}$. For each partition calculate $\frac{|\boldsymbol{S}_{P_k}|}{|\boldsymbol{S}_{P_k}|+|\boldsymbol{F}_{P_k}|}$, to give the observed proportion of successes in each partition.

7. Calculate the theoretical proportion of successes within a partition as the average of the probabilities in the combined vector of successes and failures, $(\boldsymbol{S}'_{P_k}, \boldsymbol{F}'_{P_k})'$. Here, 'theoretical', means under the assumption that the causal probabilities calculated are correct. This step and the previous are done for both the Bayesian and DET approaches. For a large number of iterations T, the theoretical proportion of each partition would be close to the midpoint of that partition.

It is then of interest to compare, within each partition, the theoretical proportion of predictors that are truly causal with the observed number of predictors that are truly causal. This gives a measure of the consistency of each method. We did this for $T = 10000$ repeats; the results are presented in Tables **??** and **??**. In each of these tables, the third column gives the difference between the observed and theoretical probabilities, and in the fourth column each difference is scaled by dividing by the theoretical probability. Both of these difference measures are summed over the 20 partitions to give a difference score. The Bayesian method does a slightly better job than DET, whichever way the total difference is measured. For both methods, however, when the causal probability is estimated as high, the predictor in question is causal less often than expected. So both methods seem to be slightly anti-conservative.

We also compare the ability of the two methods to assign high probabilities to truly causal predictors, and low probabilities to predictors with indirect effects. The average true causal probability is defined as the mean of the probabilities in the vector $\boldsymbol{S}$ for each of the Bayesian and DET stage two approaches; the average non causal probability is the mean of the probabilities in the vector $\boldsymbol{F}$, for each approach. We also include the proportion of

Table 7.1: Theoretical and observed proportions of successes within each probability partition for Bayesian method

Partition	Observed	Theoretical	Difference	Scaled diff.
P_1	0.018	0.023	0.005	0.217
P_2	0.063	0.073	0.010	0.137
P_3	0.141	0.121	0.020	0.165
P_4	0.209	0.172	0.037	0.215
P_5	0.267	0.223	0.044	0.197
P_6	0.314	0.274	0.040	0.146
P_7	0.363	0.324	0.039	0.120
P_8	0.398	0.374	0.024	0.064
P_9	0.434	0.424	0.010	0.024
P_{10}	0.492	0.474	0.018	0.038
P_{11}	0.517	0.525	0.008	0.015
P_{12}	0.554	0.575	0.021	0.037
P_{13}	0.571	0.625	0.054	0.086
P_{14}	0.617	0.674	0.057	0.085
P_{15}	0.648	0.725	0.077	0.106
P_{16}	0.697	0.775	0.078	0.101
P_{17}	0.746	0.825	0.079	0.096
P_{18}	0.785	0.875	0.090	0.103
P_{19}	0.842	0.925	0.083	0.090
P_{20}	0.918	0.979	0.061	0.062
Total			0.855	2.104

Table 7.2: Theoretical and observed proportions of successes within each probability partition for DET stage 2 method

Partition	Observed	Theoretical	Difference	Scaled diff.
P_1	0.035	0.027	0.008	0.296
P_2	0.077	0.070	0.007	0.100
P_3	0.111	0.119	0.008	0.067
P_4	0.145	0.169	0.024	0.142
P_5	0.256	0.225	0.033	0.147
P_6	0.315	0.276	0.039	0.141
P_7	0.352	0.325	0.027	0.083
P_8	0.375	0.375	0.000	0.000
P_9	0.406	0.425	0.019	0.045
P_{10}	0.431	0.475	0.044	0.093
P_{11}	0.454	0.525	0.071	0.135
P_{12}	0.490	0.574	0.084	0.146
P_{13}	0.514	0.625	0.111	0.178
P_{14}	0.550	0.675	0.125	0.185
P_{15}	0.585	0.724	0.139	0.192
P_{16}	0.632	0.774	0.142	0.183
P_{17}	0.675	0.825	0.150	0.182
P_{18}	0.751	0.874	0.123	0.141
P_{19}	0.822	0.924	0.102	0.110
P_{20}	0.921	0.972	0.051	0.052
Total			1.307	2.618

Table 7.3: Average probabilities attributed to true causal and non causal predictors. In brackets, the proportion of truly causal predictors assigned a causal probability of 0.75 or greater, and the proportion of non causal predictors assigned a probability of 0.1 or less are given.

	Bayes	DET
$\boldsymbol{S}$	0.290 (0.099)	0.316 (0.128)
$\boldsymbol{F}$	0.079 (0.780)	0.076 (0.858)

truly causal predictors assigned a causal probability of 0.75 or greater, and the proportion of non causal predictors assigned a probability of 0.1 or less. The results are given in Table **??**, and we see that, on both of these measures, DET out-performs the Bayesian approach.

Chapter 8

Conclusions and Future Directions

8.1 Label Switching and Mixture Models

Mixture models are used in situations involving heterogenous populations. They consist of a number of components, where each component represents a homogenous sub-population within the population. When conducting MCMC inference, the order that the components are in can change multiple times between iterations. This makes it difficult to carry out component-specific inference, since one cannot tell how the components in one iteration map to the components in any other iteration. This is known as the label switching problem.

To deal with the label switching problem, deterministic ‘relabelling’ strategies exist that attempt to align the components, so that essentially they are in the same order on each iteration. This process provides an estimate of how the components correspond to each other between iterations. These deterministic methods, however, provide no indication of the uncertainty that is associated with the relabelling process. We introduced a probabilistic algorithm as an alternative to the deterministic methods. Probabilistic relabelling algorithms provide a natural way to incorporate the uncertainty in the relabelling process, and have appealing connections to the EM and SEM algorithms. Applications of the approach in genetics include making inferences on sub-populations within a sample. This will become more important when larger samples, where the participants come from more diverse backgrounds, are collected, since genetic variants may have different phenotypic consequences in different sub-populations. There are many applications outside of genetics, since the mixture model is a general methodological tool. Some very recent examples of mixture model applications include longitudinal clinical trial data (?), social networks (?) and mutilevel data (?). The label switching problem is also relevant to hidden Markov models on a discrete state space (?).

An interesting avenue for future research in this area is the integration of the relabelling procedure with the determination of the number of components in the mixture model. This is suggested in Section ??. A combined approach to dealing with these two difficult issues when using mixture models would certainly be useful.

8.2 Direct Effect Testing and Sparsity Models

In the second part of the thesis we introduced a novel method for searching for direct effects in supervised problems, called direct effect testing (DET). A direct effect is a relationship between a predictor and response that is not explained by correlation with any of the other measured predictors. Any relationship between a predictor and the response that is not a direct effect is called an indirect effect. An indirect effect can arise when a predictor is correlated with another predictor that has a direct effect. DET is useful in situations where a large number of predictors are available to explain a response. For example, the response may be a disease outcome, and the predictors SNPs (single nucleotide polymorphisms). When there are more predictors than observations, or the predictors are highly correlated, or both, it is difficult to distinguish predictors directly associated with a response from those indirectly associated with the response. DET is a method that allows these situations to be handled, by distinguishing between direct effects and indirect effects, and providing information on the uncertainty in this distinction.

DET is a two stage procedure. In the first stage, lasso regression is used in a novel structure to model direct and indirect effects between the predictors and the response. This allows effects to be identified. The second stage of DET then quantifies the uncertainty in which predictors are responsible for each detected effect. Specifically, a discrete probability distribution is calculated for each detected effect, giving the probability that each predictor is directly responsible for that effect.

It may be beneficial, for future research, to consider the two stages of DET in isolation. The first stage involves fitting a lasso regression based on minimising the correlated residuals rather than the residuals. This is discussed, in Section **??**, to be a compromise between the classical lasso (**?**) and the Dantzig selector (**?**), and this is worth pursuing further, to understand the relationships between these three methods. Hypothesis testing on effect size is then carried out, based on the noncentral hypergeometric distribution of the underlying cell counts. Generalisation of this stage to continuous situations will be of interest.

The DET method was originally developed in a frequentist setting, but stage 2 of DET is making pseudo-Bayesian statements, suggesting that it would be more natural to construct the method using a Bayesian framework. The more strictly Bayesian alternative to stage two of DET, which is discussed briefly in Section **??**, seems comparable to the original proposal. The Bayesian alternative also allows more explicit inclusion of any prior knowledge that is available. Hence we believe that the Bayesian paradigm is the natural way to take the ideas of DET forward.

There are many methods that are similar to DET in that they carry out inference in the $p \gg n$ setting. Each of these methods was designed for a different purpose, but we were interested in how well these different methods recovered the 'true sparsity pattern'. The true sparsity pattern is important as it tells us which genetic variants affect our outcome of interest. A thorough simulation study showed that, surprisingly, many existing methods are good at recovering the true sparsity pattern, even in unfavourable conditions such as high correlation between predictors. The vast family of methods, including DET, have many close relationships, and more work needs to be done to understand how they all relate. From a practical point of view, a user of $p \gg n$ methods needs to know which method to use to solve their particular problem.

In genetics, identifying predictors with direct effects is potentially useful. For example, if the predictors are SNPs, genetic engineering could modify the SNP in question. There are also applications of such a procedure outside of genetics, such as digital communications (**?**), network recovery (e.g. **?**) and wavelets (**?**).