Exploring Nonlinear Regression Methods, with Application to Association Studies

Douglas Christopher Speed

St Catharine's College

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy

Department of Applied Mathematics and Theoretical Physics,
Centre for Mathematical Sciences,
Wilberforce Road,
Cambridge, CB3 0WA,
United Kingdom.

September 2010

To my Parents.

With special thanks to Simon, Marion, Christina, Radhika, Shamith
and everyone who helped me out.

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

This thesis has been typeset in 12pt font using LaTeX2ε according to the specifications defined by the Board of Graduate Studies and the Mathematics Degree Committee.

September 30, 2010

Douglas Christopher Speed
St. Catharine's College

**Exploring Nonlinear Regression Methods,
with Application to Association Studies**

The field of nonlinear regression is a long way from reaching a consensus. Once a method decides to explore nonlinear combinations of predictors, a number of questions are raised, such as what nonlinear combinations to permit and how best to search the resulting model space. Genetic Association Studies comprise an area that stands to gain greatly from the development of more sophisticated regression methods. While these studies' ability to interrogate the genome has advanced rapidly over recent years, it is thought that a lack of suitable regression tools prevents them from achieving their full potential.

I have tried to investigate the area of regression in a methodical manner. In Chapter 1, I explain the regression problem and outline existing methods. I observe that both linear and nonlinear methods can be categorised according to the restrictions enforced by their underlying model assumptions and speculate that a method with as few restrictions as possible might prove more powerful. In order to design such a method, I begin by assuming each predictor is tertiary (takes no more than three distinct values). In Chapters 2 and 3, I propose the method *Sparse Partitioning*. Its name derives from the way it searches for high scoring partitions of the predictor set, where each partition defines groups of predictors that jointly contribute towards the response. A sparsity assumption supposes most predictors belong in the "null group" indicating they have no effect on the outcome. In Chapter 4, I compare the performance of *Sparse Partitioning* to existing methods using simulated and real data. The results highlight how greatly a method's power depends on the validity of its model assumptions. For this reason, *Sparse Partitioning* appears to offer a robust alternative to current methods, as its lack of restrictions allows it to maintain power in scenarios where other methods will fail.

Sparse Partitioning relies on Markov chain Monte Carlo estimation, which limits the size of problem on which it can be used. Therefore, in Chapter 5, I propose a deterministic version of the method which, although less powerful, is not affected by convergence issues. In Chapter 6, I describe *Bayesian Projection Pursuit*, which adds spline fitting into the method to cope with non-tertiary predictors.

Glossary of Symbols

Below is an outline of the notation I use throughout this thesis.

Data Variables

- n - number of samples, indexed by the variable i.
- N - number of predictors, indexed by the variable g.
- $\boldsymbol{X}$ - predictor matrix (size $n \times N$).
- $\boldsymbol{Y}$ - response vector (length n).

Subscripts are used to indicate particular elements of a vector or matrix.
Negative subscripts indicate a vector/matrix with those elements removed.

Regression Equation

$l(\mathbb{E}(\boldsymbol{Y})) = f(\boldsymbol{X})$ - l is the link function, while f is the underlying relationship.

Partitioning Notation

- $\mathbb{G} = \{G_0, G_1, \ldots, G_K\}$ - a partitioning of (up to C copies of) {1,2,...,N}.
  The null group G_0 indexes predictors not associated,
  $G_1, \ldots, G_K$ index groups of at most S associated predictors.
- $\boldsymbol{I} = (I_1, I_2, \ldots, I_N)$ - alternative representation of $\mathbb{G}$.
  $I_g \in \{0, 1, \ldots, K\}$ denotes to which group predictor g belongs.
- $\boldsymbol{f} = \{f_1, f_2, \ldots, f_K\}$ - functions acting on each non-null group.

$S_{\boldsymbol{I}}$ lists the associated predictors: $g \in S_{\boldsymbol{I}} \Leftrightarrow I_g \neq 0$.
$[\boldsymbol{I}]$ denotes an equivalence class of partitions: $\boldsymbol{I}' \in [\boldsymbol{I}] \Leftrightarrow S_{\boldsymbol{I}'} = S_{\boldsymbol{I}}$.

The underlying relationship is modelled as $f(\boldsymbol{X}) = f_1(X_{\boldsymbol{G}_1}) + f_2(X_{\boldsymbol{G}_2}) + \cdots + f_K(X_{\boldsymbol{G}_K})$.

Projection Pursuit

- $\boldsymbol{\xi}_k = (\xi_{k1}, \xi_{k2}, \ldots, \xi_{kN})$ - direction coefficients for the kth group.
- $\boldsymbol{\Upsilon} = (\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_N)$ - condensed representation of $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_K\}$.
  Can merge $\xi_{1g}, \xi_{2g}, \ldots, \xi_{Kg} \to \Upsilon_g$, as at most one is non-zero.

The projection pursuit model is $f(\boldsymbol{X}) = f_1(\boldsymbol{X}\boldsymbol{\xi}_1) + f_2(\boldsymbol{X}\boldsymbol{\xi}_2) + \cdots + f_K(\boldsymbol{X}\boldsymbol{\xi}_K)$.

Miscellaneous

$\mathbb{P}(\square)$ denotes a probability mass/density function of the random variable $\square$.
$\square$ discrete $\Rightarrow \mathbb{P}(\square)$ is a mass function; $\square$ continuous $\Rightarrow \mathbb{P}(\square)$ is a density function.

List of Abbreviations

Below are some abbreviations and phrases I use more than once in this thesis.

Regression Methods

Single - Tests each predictor individually for association (my own implementation).
Pairs - Tests all pairs of predictors for association (my own implementation).

CART - Classification and Regression Trees (Breiman *et al.*, 1984).
RF - Random Forests (Breiman, 2004).
SSS - Shotgun Stochastic Search (Hans *et al.*, 2007).
Logic - Logic Regression (Ruczinski *et al.*, 2003).
MARS - Multivariate Adaptive Regression Splines (Friedman, 1991).

Genetic Terms

SNP - Single nucleotide polymorphism - a single base pair mutation.
LD - Linkage disequilibrium - the tendency of nearby genetic variants to exhibit strong correlations, due to being frequently inherited jointly through generations.

Mathematical Terms

MCMC - Markov chain Monte Carlo - A stochastic technique for sampling from a posterior distribution when explicit calculation is not (readily) possible.

$\delta_{\{a\}}$ - The delta function - a point mass function at a, whose integral is defined as 1.
$\mathbf{1}(\cdot)$ - The indicator function which takes value 1 if and only if its argument is true.

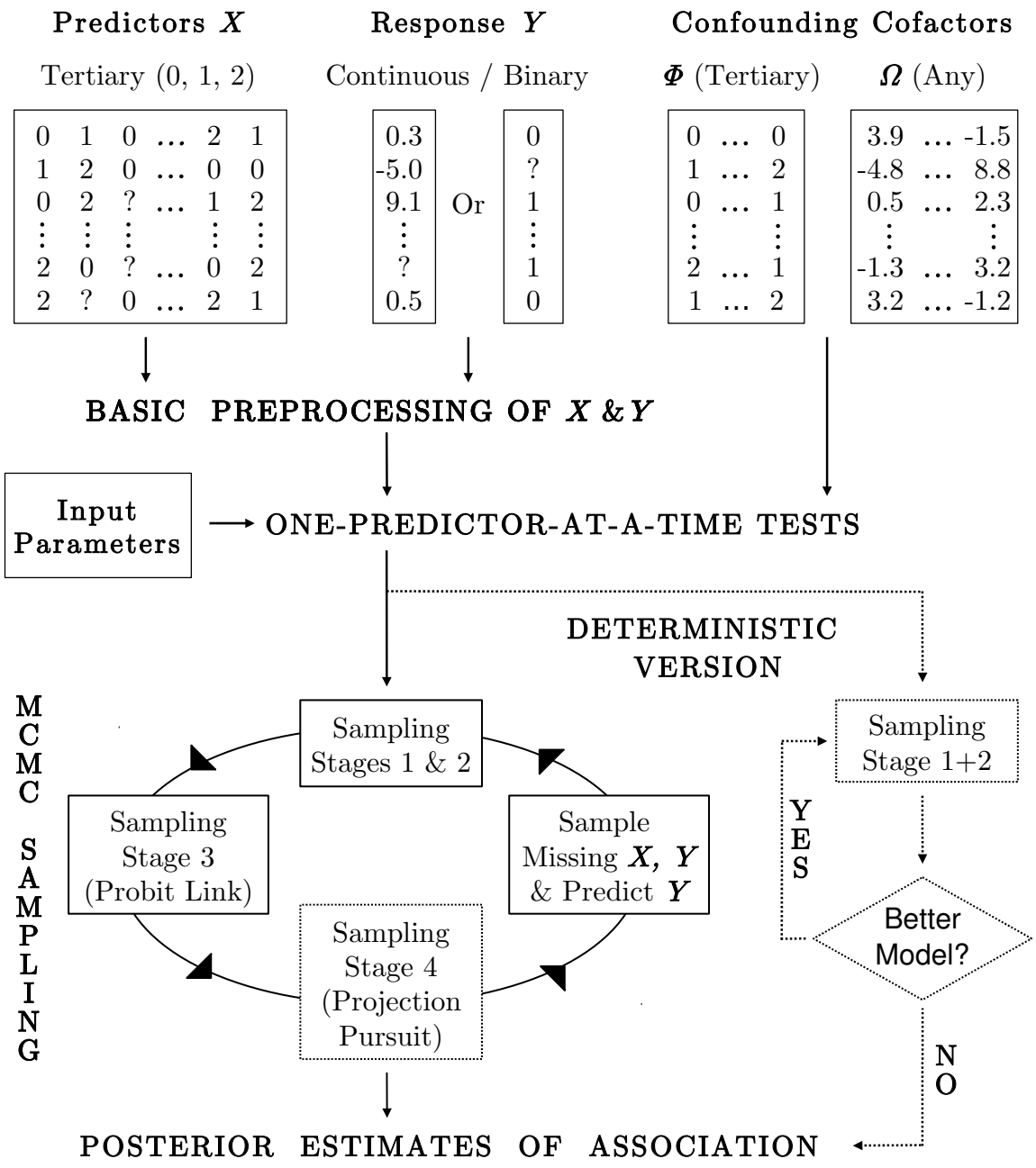
$\mathbb{U}(a, b)$ - A uniform distribution on the interval $[a, b]$.
$\beta(a, b)$ - A Beta distribution with shape parameters a and b.
$\mathbb{B}$(a,b) - The Beta function, the normalising constant of the Beta distribution $\beta(a, b)$.

$\mathbb{N}(a, b)$ - A normal distribution with mean a and variance b.
$\phi(a)$ - The probability density function of the standard normal distribution $\mathbb{N}(0, 1)$.
$\Phi(a)$ - The cumulative density function of the standard normal distribution $\mathbb{N}(0, 1)$.

Link Functions

Logit(a) - The logit function, equal to $\log(a/(1-a))$ - a commonly used link function which maps a probability to a real value.
Probit(a) - The probit link function, equal to $\Phi^{-1}(a)$, the inverse cumulative density function of the standard normal distribution - an alternative to the logit link.

Overview of Sparse Partitioning Algorithms



Three are three *Sparse Partitioning* algorithms. Solid lines indicate steps involved in the original version, dotted lines refer to the two alternatives. The original version, suitable for tertiary predictors, implements Markov chain Monte Carlo sampling. It performs Sampling Stages One, Two and Three once per iteration. The deterministic version (*Deterministic SP*) replaces the MCMC iterations with a hill-climbing search. The spline version (*Bayesian Projection Pursuit*), which can also be applied to non-tertiary predictors, carries out MCMC sampling with the addition of Sampling Stage Four.

Contents

Chapter 1

Introduction

Regression problems occur in all walks of life. Whenever we encounter an outcome whose behaviour we do not adequately understand, our instinct is to seek an explanation. The obvious first step is to look for all variables (the predictors) we think might affect the outcome (the response). If we are able to measure both the response and predictors across a sample, we have a regression problem. It is at this point statistical analysis is required, as we hope to determine how the predictors influence the response.

The field of genetics provides countless examples of this type. Perhaps the most famous concerns human height. For over 100 years, geneticists have studied the heritable nature of this trait (Galton, 1886). It is anticipated that at least 80% of the observed variation can be assigned to genetic factors, but so far the actual amount explainable falls comfortably short of this figure (Visscher, 2008). There are two possible reasons why our understanding of genetic problems is so poor: either the catalogue of variants that we have built up, which on the surface appears increasingly comprehensive, continues to overlook the true source; or our methods for analysing these data are not sufficient (Manolio *et al.*, 2009; Cordell, 2009).

In this thesis, I investigate ways to better analyse regression problems. After a brief overview of the problem, I introduce *Sparse Partitioning*, a nonlinear Bayesian method designed for predictors taking no more than three distinct values. I then discuss two extensions: *Deterministic SP*, an adaptation which removes the random component to improve speed and usability, and *Bayesian Projection Pursuit*, a version which no longer insists upon three-valued predictors. My work is heavily motivated by genetic problems, so I intersperse description with examples from association studies, but hopefully is in no way limited to this field.

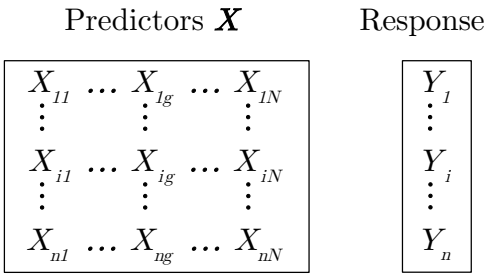


Figure 1.1: *Notation. The predictor values are stored in the matrix $\boldsymbol{X}$, the response values in the vector $\boldsymbol{Y}$. In both cases, rows indicate samples. The predictor values for the ith sample are represented by $(X_{i1}, X_{i2}, \ldots, X_{iN})$, while Y_i denotes its response.*

1.1 Regression Notation

Consider a regression problem involving n samples, N predictors and a single response. Figure 1.1 demonstrates how the data are stored. The matrix $\boldsymbol{X}$ (size $n \times N$) contains the predictors, while the column vector $\boldsymbol{Y}$ (length n) contains the response values. Throughout this thesis, the subscript i corresponds to a sample and g to a predictor. I will use notation of the type X_i or X_g to refer to the rows or columns of a matrix corresponding to particular samples or predictors. A negative subscript designates a matrix or vector with elements excluded; for example, X_{-g} denotes $\boldsymbol{X}$ with the gth column taken out, while Y_{-i} denotes $\boldsymbol{Y}$ with the ith value removed.

A predictor can be treated as either categorical or quantitative. A categorical predictor records values on a nominal scale, where its value indicates in which state the predictor occurs. I will often talk about "binary" or "tertiary" predictors. These are categorical predictors which occur in only two or three states. As the choice of labelling is of no consequence, I will assign these predictors values from $\{0, 1\}$ or $\{0, 1, 2\}$, respectively.

Quantitative predictors are ordinal; their values serve a greater purpose than simply distinguishing group membership. Continuous predictors are one such example. Suppose we are told three individuals weigh 50 kg, 51 kg and 70 kg. This provides us with more information than the fact their weights are different; it tells us that the second sample weigh more than the first, but less than the third. Based simply on these values, it would be natural to assume the first two individuals are more closely matched than the second and third.

In a similar fashion, the response can be either categorical or quantitative. The two most common situations involve either a binary or a continuous response. When binary, the labelling is again of no importance, so I will use values from $\{0, 1\}$.

A regression method is interested in deducing properties of the regression equation, the formula which connects a sample's predictors to its response. This is typically written as $l(\mathbb{E}(\boldsymbol{Y})) = f(\boldsymbol{X})$, where l is termed the link function and I refer to $f(\boldsymbol{X})$ as the "underlying

relationship". When the response is continuous, the link function is typically the identity function, so that $f(X_i)$ determines the expected response for the ith sample. When the response is binary, the link function relates $f(X_i)$ to the probability that the ith sample's response is 1, mapping a real value to the interval $[0, 1]$. In this case, either the logit or probit function (both of which are defined later on) provide a convenient choice.

With the link function specified, the task of the regression method is to identify details of the underlying relationship $f(\boldsymbol{X})$. Ideally, we wish to know the exact form of the relationship, however, details of which predictors contribute most can still prove very informative.

1.2 Association Studies

Association studies seek to answer the question "Which genetic variants affect a phenotypic trait?" There are many reasons why association studies might be of interest. To provide just two, first suppose the phenotype is disease-based. If we are able to understand the biological system underlying this response, it will hopefully result in better preventative measures and allow more specialised treatment. If instead the phenotype measures crop performance, understanding what causes some plants to thrive more than others should suggest ways to increase overall yield.

Reworded as a regression problem, each study asks "Which predictors are associated with the response?" An association study's first step is to select a set of samples. If the phenotype to be investigated has already been decided, it is natural to choose samples providing a broad spectrum of response values, as these will be expected to highlight causal variants most clearly. The second step is to type each sample for a number of predictors. Over the past decade, there has been a rapid progression in the ability to record genomic variants, both in terms of the types of variants explored (Manolio *et al.*, 2008) and the density at which they can be measured (The International HapMap Consortium, 2003, 2004, 2007).

When comparing two genomes, one commonly considered variant is the "SNP" (single nucleotide polymorphism). A SNP is known to exist once more than one base pair value has been observed at a particular location. Owing to the vastness of DNA and the rarity of mutations, SNPs are generally considered "biallelic", meaning that the location assumes one of only two states ("alleles"). On a population level, it is often required that this variation is sufficiently wide-spread before a SNP is declared. For example, it is customary to insist the "minor allele frequency", the proportion at which the least common allele is observed, is at least 1%. For many species, chromosomes occur in closely matched ("homologous") pairs. As chromosomes within a pair are hard to distinguish, a SNP value is generally recorded by how often the minor

allele occurs across both, so equals 0, 1 or 2, depending on whether the sample is homozygous wildtype, heterozygous, or homozygous mutant.

As a child's DNA is a composition of its parents', it would, in theory, be possible to construct a tree detailing the origin of every sequence position in the current generation. Each ancestral allele could be traced back to the founder generation, while each mutant allele could be traced back to the time it first appeared. The manner in which alleles are passed through generations is far from random, and modelling this process forms the basis of coalescent theory (Hein *et al.*, 2005). In particular, neighbouring base pairs will very often originate from the same parent, resulting in high concordance between nearby variants. On a population level, this phenomenon is referred to as "linkage disequilibrium" (LD) and is recognised by local patterns of strong correlation between groups of predictors (Nordborg and Tavaré, 2002). Association studies are able to exploit LD to reduce the experimental workload. For example, the HapMap Project estimates there to be of the order 10 million SNPs in the human genome, but because of the strong correlations present, much of the variation can be captured by genotyping a much smaller subset of these (Conrad *et al.*, 2006). Having typed a set of "tagging" SNPs, a study can then choose either to analyse this subset directly, or impute untyped variants using reference genomes (Marchini and Howie, 2010).

Association studies vary greatly in scale, depending on the density of variants and the length of genomic sequence typed. With current technology, it is common-place for whole genome studies to interrogate up to a million SNPs (Psychiatric GWAS Consortium Coordinating Committee, 2009), yet an experiment with a more focused objective might concentrate on less than a few hundred. Either way, the majority of studies are classed "large p, small n" problems, an expression used to describe regression problems where the number of predictors far exceeds the number of samples. This has implications in their analysis. With access to enough predictors, it will be possible to find models which perfectly explain any set of observed response values, but there is no guarantee these models are meaningful.

The heritability of a trait represents the proportion of phenotypic variation which can be attributed to genetic effects. At one extreme are Mendelian traits, named after Gregor Mendel, an Austrian monk whose experiments were fundamental to their understanding (Mendel, 1865). For these traits, the presence/absence of (typically) one causal allele will explain 100% of the observed variation. For example, much of Mendel's work concerned *Pisum sativum*, the seed colour of which is controlled by a single genomic location. If at least one copy of the "dominant" allele is present across the homologous pair, the seed is yellow; whereas if two copies of the "recessive" allele are present, the seed is green. The alleles underlying Mendelian traits should be fairly easy to detect; assuming the causal variant has been typed (or well-tagged), one simply has to look for the predictor whose values correlate perfectly (or very

highly) with the response. The fact that the majority of phenotypes can not so easily be explained, strongly suggests that their underlying systems are far more complex, with more than one variant causal and/or a greatly reduced heritability.

Following the acceptance that most traits are unlikely to be Mendelian, much attention became focused on the "common disease, common variant" hypothesis. This supposes that the genetic variation underlying many phenotypes can be attributed to a relatively small number of reasonably common variants. While association studies were in their infancy, this hypothesis agreed with many of the known findings (mainly discovered through family-based studies) and with the genetic models being proposed (Risch and Merikangas, 1996; Kruglyak, 1999; Reich and Lander, 2001). However, in recent times, this hypothesis has been called into question. Although there have been a number of successes, most notably those of The Wellcome Trust Case Control Consortium (2007, 2010), there remain many examples, like that of human height, of highly heritable traits for which association studies have not lived up to expectations.

There has been much speculation as to why this should be the case (Maher, 2008; Goldstein, 2008; McCarthy *et al.*, 2008). One conclusion is that the causal variants are more abundant and/or much rarer than was first thought. On the other hand, the success of a study will always be limited by the efficiency of its analysis. In particular, the majority of analysis methods assume an additive model, however, there has been evidence for interactions affecting a phenotypic outcome, in animals at least (Fraser, 2007; Shao *et al.*, 2008), thus raising the question of whether the current approaches are sufficient (Balding, 2006; Thomas, 2010).

1.3 Linear Models

An underlying relationship f is classed as linear, with respect to $J_1, J_2, \ldots, J_D$, if it can be written as a linear combination of these predictors:

$$f(J_1, J_2, \ldots, J_D) = J_1\theta_1 + J_2\theta_2 + \ldots + J_D\theta_D,$$

where D is referred to as the degrees of freedom, the minimum number of parameters required to describe the model. If we create $\boldsymbol{J} = [J_1\ J_2 \cdots J_D]$, a design matrix whose columns are the predictor values, and a vector of regression coefficients $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_D)^T$, then the linear model can be written as $f(\boldsymbol{J}) = \boldsymbol{J\theta}$.

If $\{J_1, J_2, \ldots, J_D\}$ is a subset of $\{X_1, X_2, \ldots, X_N\}$, then the underlying relationship is also a linear function of the original predictors. But this is by no means required, as linear models also play a part in nonlinear regression. Suppose we let $J_1 = X_1 \times X_2$, which for binary X_1 and

X_2 indicates whether or not both equal one. By considering linear models involving J_1, we are able to consider nonlinear models with respect to the original predictors. This is a strategy I will use repeatedly, using indicator matrices to construct nonlinear functions of columns of $\boldsymbol{X}$.

Given a linear model involving $\boldsymbol{J}$, we are interested in finding the most suitable values for the regression coefficients. In terms of the regression equation, this corresponds to finding $\hat{\boldsymbol{\theta}}$ such that $f(\boldsymbol{J}) = \boldsymbol{J}\hat{\boldsymbol{\theta}}$ is the "best fit" to $l(\mathbb{E}(\boldsymbol{Y}))$. How we decide upon the best fitting $\hat{\boldsymbol{\theta}}$ is an entirely subjective choice, dependent on our aversion to error and insights regarding the regression coefficients. When the response is continuous, by far the most common frequentist strategy is least squares regression, which picks $\boldsymbol{\theta}$ to minimise

$$(\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\theta})^T(\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\theta}) = \sum_i (Y_i - J_i\boldsymbol{\theta})^2.$$

This expression represents the sum of the squares of the residuals, the difference between the predicted and observed values for each response. In many cases, the least squares estimate $\hat{\boldsymbol{\theta}}$ can be calculated explicitly as

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{Y}.$$

This relies on the matrix $\boldsymbol{J}^T\boldsymbol{J}$ being invertible, which in turn relies on linear independence of the observed predictor values. Put simply, if two predictors are identical, for example, $J_1 = J_2$, the model lacks identifiability, as increasing θ_1 while decreasing θ_2 by the same amount will have no change on the underlying relationship. More formally, this phenomenon will exist whenever the predictor set is linearly dependent, as then one predictor can be expressed as a linear combination of the others.

This concept is closely related to the idea of saturation. For a sample of size n, there can be at most n linearly independent predictors. Once the number of linearly independent predictors equals the number of samples, the model is termed saturated and a perfect fit will always be achievable. As a result, adding further predictors to the linear model will have no effect on the goodness of fit, while the lack of identifiability means there will be no unique solution.

It is often possible to avoid problems of saturation by introducing a penalty term. For example, rather than finding the least squares estimate, we could consider minimising the penalised residual sum of squares

$$(\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\theta})^T(\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\theta}) + \text{Pen}(\boldsymbol{\theta}).$$

The scalar penalty function $\text{Pen}(\boldsymbol{\theta})$ generates additional constraints and often allows models

with more than n degrees of freedom to be uniquely solved. This penalty term will be picked to reward models with desirable properties, so typically reflects a preference for simplicity.

Least squares estimation ties in nicely with maximum likelihood statistics. A common assumption is that the residuals are independent draws from a normal distribution, in which case their likelihood, the probability function corresponding to a set of observed values, takes the form

$$(2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y}-\boldsymbol{J\theta})^T(\boldsymbol{Y}-\boldsymbol{J\theta})\right\}.$$

The choice of $\boldsymbol{\theta}$ which maximises the likelihood will be that which minimises the exponent. Therefore, the maximum likelihood and least squares estimates are the same.

1.3.1 Significance Tests

Having found a best fit, we often wish to attach significance to this finding. For example, we might like to investigate how much evidence there is that a regression coefficient is non-zero, indicating that the corresponding predictor has an effect on the response. A frequentist solution is to perform a likelihood ratio test. When the null hypothesis is nested within the alternative, this test calculates the statistic

$$\Lambda = -2\log\frac{L(\hat{\boldsymbol{\theta}}_{\mathbf{0}})}{L(\hat{\boldsymbol{\theta}}_{\mathbf{1}})},$$

where $\hat{\boldsymbol{\theta}}_{\mathbf{0}}$ and $\hat{\boldsymbol{\theta}}_{\mathbf{1}}$ correspond to the best fitting models under the null and alternative hypotheses. The greater the value of Λ, the more evidence there is to reject the null hypothesis. For the test statistic, we generally wish to calculate a "p-value", which represents the probability under the null hypothesis of obtaining a value at least as extreme as that seen. This requires knowledge of the distribution of Λ when the null hypothesis is true. If this distribution can not be calculated directly, it can often be approximated through use of an asymptotic argument.

Suppose a significance test leads to the rejection of the null hypothesis. This result will be a "true positive" if the right decision was made, but a "false positive" when the null hypothesis was in fact correct. Typically, we will choose a significance level α' and reject the null hypothesis if the p-value falls below this level. The "power" of a test, for a particular significance level, is the probability it will correctly reject, should the null hypothesis be false.

1.3.2 Multiple Testing

When performing a number of significance tests, it is common to consider the "family-wide error rate" (FWER), the chance of incorrectly rejecting one or more null hypotheses. If we perform T independent tests, each with a significance level of α', the FWER will equal $1-(1-\alpha')^T$.

Therefore, if we wish to set the FWER to α, it is straightforward to calculate the required value for α' (Šidák, 1967).

Problems arise when the tests are not independent. Boole's Inequality states that the probability of one or more events happening is no greater than the sum of the events' individual probabilities. Based on this fact, Bonferroni correction suggests we set $\alpha' = \alpha/T$, as then the FWER can not exceed α. However, this approach will often prove overly conservative. Consider an extreme case, when we perform the same test for two identical predictors. Each test will calculate the same test statistic, make the same distributional assumptions and therefore produce the same p-value and outcome. If we obeyed Bonferroni correction and set $\alpha' = \alpha/2$, the FWER would actually be half what we desired. One approximate solution might be to estimate the effective number of independent tests (which in this simple case would be 1) and perform Bonferroni correction using this value in place of T.

Permutation testing provides an alternative means of assessing significance, one which requires no knowledge of the null distribution of the test statistic. Suppose our null hypothesis states that there is no true association between the response and a predictor. We can replicate this situation by permuting the response values, as then any observed association will have occurred purely through chance. Therefore, any test statistic calculated on the permuted data will represent a draw under the null hypothesis; with sufficient permutations, we can obtain a p-value by comparing these draws with the test statistic calculated for the actual data.

Permutation testing is suitable even when the tests are correlated. Suppose we wish to test each of T predictors for association with a response. If we record, for each permutation, the collection of T test statistics, we can estimate the joint distribution of these statistics under the null hypothesis. This then allows us to calculate a suitable threshold, one which takes into account any possible dependencies between tests. The drawbacks of permutation testing stem from their computational burden. The resolution of the p-value obtained is constrained by the number of permutations, but each permutation requires the entire analysis be repeated. For example, to be able to declare a result significant at a 0.001 threshold requires at least 1000 permutations, and many more if we desire reasonable certainty. By approximating the shape of the tail of the null distribution, Knijnenburg *et al.* (2009) examine ways to increase the accuracy of extreme p-values obtained through permutation testing. But while they demonstrate the resolution of very small values can be increased by upwards of 3 orders of magnitude, to get to this point a minimum of a few thousand permutations are still required.

In some situations, we may no longer be interested in controlling the FWER. Suppose we are testing very many predictors-response pairs and expect there to be a reasonable number of true associations. In this case, we might be content to declare a few incorrect associations

provided the majority of our declarations are accurate. The proportion of declarations which are incorrect is known as the false discovery rate (FDR; Benjamini and Hochberg, 1995). Allowing a higher FDR indicates that we are willing to sacrifice specificity (allow more false positives), in the hope of greater sensitivity (more true positives).

1.3.3 The Bayesian Approach

Frequentist methods base parameter inferences almost exclusively on the evidence provided by the data. By contrast, Bayesian methods incorporate prior knowledge as well. Bayesian methods are concerned with the evaluation of the posterior distribution of the parameters $\mathbb{P}(\text{Parameters}|\text{Data})$, at the centre of which is Bayes' formula:

$$\mathbb{P}(\text{Parameters}|\text{Data}) \propto \mathbb{P}(\text{Data}|\text{Parameters}) \times \mathbb{P}(\text{Parameters}).$$

Here $\mathbb{P}(\text{Data}|\text{Parameters})$ is the likelihood of the data, while $\mathbb{P}(\text{Parameters})$ is called the prior distribution. As the equation shows, the posterior distribution provides a compromise between the evidence offered by the data and the beliefs held by the prior.

Finding the maximum likelihood estimate corresponds to finding the mode of the posterior distribution when the prior is uniform. The use of more informative priors, i.e. ones which reflect a preference for certain models, can be compared to the introduction of penalty terms in the frequentist set-up. For example, consider the linear model, again treating the residuals as draws from a normal distribution, and assign independent, identically distributed, normal priors, with mean zero and variance σ^2/r, to each element of θ. The equation for the posterior distribution of the coefficient takes the form

$$\mathbb{P}(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Y}) \propto \exp\left\{-\tfrac{1}{\sigma^2}(\boldsymbol{Y} - \boldsymbol{J\theta})^T(\boldsymbol{Y} - \boldsymbol{J\theta})\right\} \times \exp\left\{-\tfrac{r}{\sigma^2}\boldsymbol{\theta}^T\boldsymbol{\theta}\right\}.$$

Finding the posterior mode for $\boldsymbol{\theta}$ equates to minimising

$$(\boldsymbol{Y} - \boldsymbol{J\theta})^T(\boldsymbol{Y} - \boldsymbol{J\theta}) + r\boldsymbol{\theta}^T\boldsymbol{\theta},$$

which, in the frequentist setting, corresponds to least squares regression with the addition of a penalty term based on the sum of squares of the regression coefficients.

The Bayesian analogy to the likelihood ratio test involves calculation of a Bayes factor (Jeffreys, 1935):

$$\text{BF} = \frac{\mathbb{P}(\text{Data}|\text{Model 1})}{\mathbb{P}(\text{Data}|\text{Model 0})},$$

where Models 0 and 1 represent the null and alternative hypotheses. The Bayes factor measures

the relative evidence for the observed data under each hypothesis; higher Bayes factors indicate greater support for the alternative. Equivalently, a Bayes factor can be expressed in terms of posterior and prior odds:

$$\text{BF} = \frac{\mathbb{P}(\text{Model 1}|\text{Data})/\mathbb{P}(\text{Model 1})}{\mathbb{P}(\text{Model 0}|\text{Data})/\mathbb{P}(\text{Model 0})} = \frac{\text{Posterior Odds}}{\text{Prior Odds}},$$

and it is often reasonable to use just the posterior odds in its place. One advantage of the Bayes factor is its ability to consider strength of evidence for the alternative hypothesis, whereas the likelihood ratio test focuses on finding evidence against the null. Saying this, in many cases, the two tests produce very similar results. For example, Wakefield (2009) formally demonstrates how, when testing for association between SNPs and a binary response, a particular choice of prior will result in p-values and Bayes factors having identical rankings.

Even when their results are comparable, I personally feel the Bayesian approach is more elegant and better equipped to cope in a range of situations. One example concerns "forwards regression" methods. These begin with an empty model, then sequentially add predictors until the decision is taken to stop. Generally the model fit will improve with each addition, so if based on fit alone, these methods would continue until saturation. Therefore, the frequentist solution is to introduce a penalty term to offset the inevitable improvement in fit. However, this term can appear quite arbitrary and lack interpretation.

In the Bayesian setting, we are able to base moves within the model space on posterior probabilities. Explicit penalty terms are no longer required, as the prior distribution of the parameters assumes this role. Quite often, our prior belief concerning a predictor's role in the underlying relationship will be two-part. Consider a linear model, in which the contribution of X_g to $f(\boldsymbol{X})$ is determined by the coefficient θ_g. When deciding upon a prior for θ_g, we might begin by estimating the probability that it is non-zero, indicating that X_g is in some way involved. Given that X_g *is* in some way involved, we can then consider a suitable distribution for its values. Here, "spike and slab" priors prove very useful, providing a formal way to represent this information. A prior of this type consists of a mixture of a point mass at zero (the spike) combined with a second distribution (the slab). The weighting given to the point mass reflects our belief that the predictor is involved; while, given that it is, the second distribution reflects our belief in its contribution.

1.3.4 Estimating the Posterior Distribution

Ideally, Bayesian methods will be able to calculate the required posterior distribution explicitly. In practice, this is often not possible. Consider a problem using spike and slab type priors. These priors "discretise" the space of possible models based on whether or not each predictor

contributes to $f(\boldsymbol{X})$. If we are given a vector from $\{0,1\}^N$, whose elements indicate which components of θ_g are non-zero, it might be possible to calculate the posterior distribution for all models corresponding to this vector. However, as we wish to find the posterior across all possible states, it will be necessary to evaluate for all 2^N vectors. Except for the most simple problems, this will not be possible.

Markov Chain Monte Carlo (MCMC) is often used for approximating the posterior distribution in these circumstances. In this case, the Markov chain represents a sequence of states within the model space, defined by the probabilities of moving from one state to another. A key property is that these probabilities depend only on the current state. If $M^1, M^2, \ldots, M^t$ represent the first t states of the chain, then state $t+1$ will be picked according to

$$\mathbb{P}(M^{t+1}|M^1, M^2, \ldots, M^t) = \mathbb{P}(M^{t+1}|M^t).$$

The premise of MCMC is to define move probabilities so that sampling from the Markov chain corresponds to sampling from the posterior distribution being sought. A Markov chain displays "detailed balance" for the distribution π if, for any two states M_1 and M_2,

$$\pi(M_1) \times \mathbb{P}(M_2|M_1) = \pi(M_2) \times \mathbb{P}(M_1|M_2).$$

Should this property hold, then π represents the chain's "stationary distribution", the marginal probabilities of each state occurring. This fact can be appreciated by considering the different ways the chain can arrive at state M_2:

$$\sum_{M_1} \big(\pi(M_1) \times \mathbb{P}(M_2|M_1)\big) = \pi(M_2) \times \sum_{M_1} \mathbb{P}(M_1|M_2) = \pi(M_2).$$

If a chain is "irreducible", meaning that there is a non-zero probability of moving between any two states within a finite number of steps, then its stationary distribution will be unique (Gilks *et al.*, 1996).

Metropolis-Hastings theory (Hastings, 1970) simplifies the task of creating a chain whose stationary distribution matches the posterior distribution. Suppose that, while at state M_1, we propose a new state M_2 with probability $\mathbb{Q}(M_2|M_1)$. Metropolis-Hastings dictates that we should accept this proposal with probability $\min(1, \alpha_{M_2|M_1})$, where

$$\alpha_{M_2|M_1} = \frac{\mathbb{P}(M_2|\text{Data})}{\mathbb{P}(M_1|\text{Data})} \times \frac{\mathbb{Q}(M_1|M_2)}{\mathbb{Q}(M_2|M_1)}.$$

If we obey this acceptance probability, we see that

$$\begin{aligned}\mathbb{P}(M_1|\text{Data}) \times \mathbb{P}(M_2|M_1) &= \mathbb{P}(M_1|\text{Data}) \times \mathbb{Q}(M_2|M_1) \times \min(1, \alpha_{M_2|M_1}) \\ &= \min\left(\mathbb{P}(M_1|\text{Data}) \times \mathbb{Q}(M_2|M_1), \mathbb{P}(M_2|\text{Data}) \times \mathbb{Q}(M_1|M_2)\right) \\ &= \mathbb{P}(M_2|\text{Data}) \times \mathbb{Q}(M_1|M_2) \times \min(1, \alpha_{M_1|M_2}) \\ &= \mathbb{P}(M_2|\text{Data}) \times \mathbb{P}(M_1|M_2).\end{aligned}$$

Therefore, the resulting chain displays detailed balance, so its stationary distribution is the posterior distribution of models, as required. We are permitted to use more than one proposal distribution, provided we obey the appropriate acceptance probability for each. Single-update Metropolis-Hastings does not move directly from M^t to M^{t+1}, but instead breaks this down into a number of steps, each of which proposes a change to one component of M^t. If the proposal distribution for each component is picked to equal that component's conditional posterior distribution, then the acceptance probability will always equal one and the move will always be accepted. This case is referred to as Gibbs' sampling.

1.4 Existing Regression Methods

I now describe some methods currently available for analysing high dimensional regression problems. In anticipation of the next chapter, consider writing the underlying relationship as a sum of functions of groups of predictors:

$$f(\boldsymbol{X}) = f_1(X_{G_{11}}, \ldots, X_{G_{1s_1}}) + f_2(X_{G_{21}}, \ldots, X_{G_{2s_2}}) + \cdots + f_K(X_{G_{K1}}, \ldots, X_{G_{Ks_K}}),$$

where s_k indicates the number of predictors contributing to f_k. Under this representation, $f(\boldsymbol{X})$ is influenced by additive contributions from groups of "interacting" predictors. There is no requirement that contributing predictors appear in only one group, however, for the existing methods, this is usually the case. Throughout this thesis, I consider two predictors to interact if their joint contribution to the underlying relationship can not be described by an additive model. For example, if the true underlying relationship takes the form $f(\boldsymbol{X}) = f_1(X_g, X_{g'})$, this implies it can not be written as $f(\boldsymbol{X}) = f_1(X_g) + f_2(X_{g'})$. As a result, I consider predictors in each group to interact with each other, but not to interact with predictors in different groups. Importantly, this representation incurs no loss of generality, as it includes the most complicated model possible, when all N predictors feature in a single group.

1.4.1 Sparsity Assumption

Each regression method explores a subspace of all possible underlying relationships. Its choice of subspace will be influenced by a combination of computational issues and intuition concern-

	ONE GROUP OF PREDICTORS	MULTIPLE GROUPS OF PREDICTORS
LINEAR	$\boldsymbol{Y} = f_1(X_{G_{11}})$ e.g. *Single*	$\boldsymbol{Y} = f_1(X_{G_{11}}) + \cdots + f_K(X_{G_{K1}})$ e.g. *SSS*
NONLINEAR	$\boldsymbol{Y} = f(X_{G_{11}}, \ldots, X_{G_{1s_1}})$ e.g. *Pairs*, *CART*, *RF*	$\boldsymbol{Y} = f(X_{G_{11}}, \ldots, X_{G_{1s_1}}) + \cdots + f(X_{G_{K1}}, \ldots, X_{G_{Ks_K}})$ e.g. *Logic*, *MARS* and *Sparse Partitioning*

Figure 1.2: *Classification of regression methods. I have categorised methods according to two features of their underlying relationships: whether they permit more than one group of predictors and whether they permit more than one predictor in each group. This table shows the four possibilities and lists some methods in each category.* Single*,* Pairs *and* Sparse Partitioning *are my own implementations, while* CART*,* RF*,* SSS*,* Logic *and* MARS *refer to existing methods, all of which I describe in the main text.*

ing the form of the true underlying relationship. In large p, small n problems, it is common to apply a sparsity assumption, one which supposes that only a small number of predictors are causal. This assumption might seem debatable. In the context of association studies, it is in line with the common disease, common variant hypothesis, the validity of which has been questioned. Similarly, from speaking to members of the Nordborg Lab, who concentrate on *Arabidopsis thaliana*, they are coming around to the idea that some traits may be affected by many tens of causalities, many of which have only a tiny effect on the phenotype.

Fortunately, I feel that, in some sense, the validity of this sparsity assumption is irrelevant. If it is the case that vast numbers of causal predictors do influence a response, there is no hope of identifying them all for standard sample sizes, so an assumption of this nature is necessary. Perhaps more accurately, however, the assumption can be worded as a prior belief in the number of "strong associations". Therefore, when I discuss "the search for associations", this phase can be interchanged with "the search for strong associations", depending on one's point of view.

A regression method can be classed as linear or nonlinear, depending on whether or not it permits interactions between predictors. In terms of my expression for the underlying relationship, this very closely corresponds to whether the method allows only one, or more than one, predictor in each group. In a similar fashion, a method can be classified based on whether it permits only one, or more than one, group of predictors. This creates four possible categories, as demonstrated in Figure 1.2.

When discussing the methods in each category, I focus mainly on those suitable for cate-

gorical predictors. There are areas, most notably that of functional data analysis, which are devoted to regression with quantitative predictors. Typically, these are designed for small numbers of predictors, and their focus is on prediction (explicitly calculating $f(\boldsymbol{X})$), rather than variable selection (identifying the groupings). Methods in this area often opt for spline fitting, a topic I discuss more in Chapter 6.

1.4.2 One Group, Maximum Group Size One

$$f(\boldsymbol{X}) = f_1(X_{G_{11}})$$

The simplest assumption supposes that the underlying relationship, and therefore the response, is influenced by only one predictor. If predictor g takes only two values, any non-trivial function of X_g will have two degrees of freedom and can be written as $f_1(X_{ig}) = \theta_{1X_{ig}}$. If X_g takes more than two values, it is necessary to decide whether to treat these values as categorical or quantitative. If categorical, the function is again a mapping of distinct points, taking the form $f_1(X_{ig}) = \theta_{1d}$, where d indicates to which image the state X_{ig} is assigned. This can be written as the linear model $f_1(X_g) = \boldsymbol{J}_1\boldsymbol{\theta}_1$, where the dth column of $\boldsymbol{J}_1$ indicates which samples are mapped to θ_{1d}. The most general form for $f_1(X_g)$ has degrees of freedom equal to the number of distinct values for X_g. However, we can also consider forms with less degrees of freedom, insisting that distinct predictor states are mapped to the same image: $f_1(X_{ig}) = f_1(X_{i'g})$ for some $X_{ig} \neq X_{i'g}$.

If quantitative, the function can be viewed as a curve, whose domain includes the possible predictor values. In theory, there is no limit to the degrees of freedom of this curve; in practice, when the degrees of freedom exceeds the number of unique values observed for X_g, there will be some redundancies. For genetic applications, it is very common to use the simplest non-trivial curve, a straight line, in which case the model takes the form $f_1(X_{ig}) = \theta_{10} + \theta_{11}X_{ig}$. Here, the intercept term θ_{10} represents the baseline value, while the gradient θ_{11} indicates how much each unit change in X_g affects the underlying relationship. Again, this is easily written as $f_1(X_g) = \boldsymbol{J}_1\boldsymbol{\theta}_1$, by letting $\boldsymbol{J}_1 = [\mathbf{1}\ X_g]$, where $\mathbf{1}$ is a vector of ones.

As there are only N choices for the causal predictor, it is straightforward to explore all possibilities. Having decided on a functional form, most frequentist methods in this category are equivalent to performing a maximum likelihood test for each predictor, comparing the null hypothesis, $f(\boldsymbol{X}) =$ constant, with the alternative, $f(\boldsymbol{X}) = f_1(X_g)$. It is easy to create a Bayesian counterpart (e.g. Balding, 2006), which instead calculates the Bayes factor or posterior probabilities for the null and alternative hypotheses. Such a version is useful when we have prior knowledge of how likely it is that each predictor is associated.

The ability of these methods to detect an associated predictor depends on the strength of

its marginal effect. For example, when testing whether a binary predictor X_g is associated, the methods will compare the response values when $X_g = 0$ to those when $X_g = 1$. Discordance between these two sets of values provides evidence that X_g has an effect on the response. Naturally, these methods perform best when the true underlying relationship actually is affected by just one predictor. When this is not the case, the presence of additional causal predictors will generally diminish each association's marginal effect and so these methods' power to detect.

These methods have proven very popular for the analysis of association study data. This owes much to their simplicity; they only require N comparisons, so computation time is kept to a minimum, and their conclusions can be explained to someone with only a basic understanding of statistics. Furthermore, considering how simple their underlying relationship assumptions, these methods have been surprisingly successful. In association studies, many hundreds of causal variants have been identified using these "one-predictor-at-a-time" approaches, some of the most high-profile finds coming from The Wellcome Trust Case Control Consortium (2007, 2010). For these reasons, methods of this type typically form the starting point for any analysis.

Single is my implementation of a method in this category, offering both a frequentist and Bayesian analysis. For the latter, given a prior probability of association for each predictor, it calculates a posterior probability of association. For the frequentist version, *Single* performs a maximum likelihood test and returns a p-value. I explain this method in more detail in Chapter 4, in particular showing the similarity between posterior probabilities and p-values when a uniform prior is used.

1.4.3 One Group, Maximum Group Size Greater Than One

$$f(\boldsymbol{X}) = f_1(X_{G_{11}}, \ldots, X_{G_{1s_1}})$$

When $s_1 = 2$, there are $\binom{N}{2}$ choices for the pair of interacting predictors and it will generally remain feasible to test all possibilities. In the case of very high-dimensional problems, which might have upwards of 500,000 predictors, an implementation of such a search may take many hundreds of computing hours (cf. Marchini *et al.*, 2005), although this can be offset with parallelisation.

When the predictors are categorical, the underlying relationship will assume the form $f_1(X_{ig}, X_{ig'}) = \theta_{1d}$, where d indicates to which image the vector $(X_{ig}, X_{ig'})$ relates. The number of unique values of d determines the degrees of freedom, and therefore the flexibility, of the model. Again, this is readily represented as a linear model, by constructing the matrix $\boldsymbol{J}_1$ whose columns indicate to which image each sample is mapped. When the predictors

are quantitative, there is no obvious choice for the functional form. One possibility might be $f_1(X_g, X_{g'}) = f_{11}(X_g) + f_{12}(X_{g'}) + f_{13}(X_g, X_{g'})$, where f_{11} and f_{12} represent the additive contributions of X_g and $X_{g'}$, while f_{13} tries to capture the "interaction term".

Pairs is my implementation of a method in this category. Designed for categorical predictors, it simply extends *Single* to additionally consider alternative models of the form $f_1(X_{ig}, X_{ig'}) = \theta_{1d}$, with full degrees of freedom. For each possible alternative model, it returns a p-value and posterior probability by comparing this to the null model $f(\boldsymbol{X}) = \text{constant}$. It is easy to imagine extending this method further, to exhaustively try all three or four-way interactions, but for all except the smallest problems, such a method would typically take far too long.

Classification and Regression Trees (*CART*; Breiman *et al.*, 1984) is another method in this category, one which can be applied to both categorical and quantitative predictors. *CART* explores the space of decision trees, each of which defines a partitioning of the samples. Within a decision tree, each internal node divides a set of samples into two groups based on the value at a specified predictor. For example, the samples with $X_{ig} \leq 1$ might be directed into one group, those with $X_{ig} > 1$ into the other. Each decision tree is scored based on its ability to explain the observed response values; a tree will score highly if its partitioning groups samples with similar response values. Once again, each model can also be written in the form $f(\boldsymbol{X}) = \boldsymbol{J}_1\boldsymbol{\theta}_1$. In this case, the dth column of $\boldsymbol{J}_1$ indicates which samples are assigned to the dth group of the partition.

CART implements a forwards regression search. At each step, it decides whether to add a predictor into the current model, which equates to adding a node to the current decision tree. As each additional predictor will generally improve the model fit, it is common to introduce a penalty term to control the model's growth. An alternative is to let the search continue until no further improvement is possible, which might well result in a tree that assigns each sample to its own group and therefore has perfect fit. At this point, the tree can be "pruned" to the desired size by removing predictors, in a process known as "backwards regression".

A noticeable difference between *CART* and *Pairs* is that the former does not insist on the full interaction model for associated predictors. For example, suppose for binary predictors *CART* decides to split the samples first based on whether $X_g = 1$, then splits the samples for which $X_g = 1$ by their value at $X_{g'}$. This model will produce three groupings so have three degrees of freedom, even though four unique vector values of $(X_g, X_{g'})$ might be present.

Random Forests (Breiman, 2004) offers a stochastic interpretation of *CART*. Out of the n samples, let n_0 represent the "training set", while the remainder form the "test set". At each

iteration, a decision tree is constructed using n_0 samples picked with replacement from the training set ("a bootstrap sample"). Each node of this decision tree is determined by choosing $N_0 << N$ predictors at random from all those available, and selecting the one which provides the best improvement in fit. Nodes are added until no further improvement is possible. Having grown a number of trees in this way, each can be scored according to its prediction accuracy for the samples in the test set. Finally, an importance weighting is calculated for each predictor by averaging the scores of all trees in which it appears. *RF* offers a practical implementation of *CART* for very large numbers of samples. The choice of N_0 serves as the penalty term, and can be interpreted as the prior belief in the correct number of associations. *RF* does not draw conclusions based on a single best fitting model, but instead calculates a weighted average over a number of models. The idea of model averaging, (discussed by HOETING *et al.*, 1999; WASSSERMAN, 2000, among others) is one which, as I will show later on, seems generally a better strategy.

1.4.4 More Than One Group, Maximum Group Size One

$$f(\boldsymbol{X}) = f_1(X_{G_{11}}) + f_2(X_{G_{21}}) + \cdots + f_K(X_{G_{K1}})$$

This underlying relationship allows more than one predictor to be causal, but insists that the causal predictors contribute independently and additively. When we introduce more than one function, it may be necessary to safeguard against unidentifiability. For example, if we have two functions, each of the form $f_k(X_{ig}) = \theta_{k0} + \theta_{k1}X_{ig}$, it is possible to alter θ_{10} and θ_{20} without changing $f_1 + f_2$. The easiest solution to this problem is to merge each θ_{k0} into a global intercept θ_0. In this case, the degrees of freedom of the model reduces from $2K$ to $1 + K$. Similarly, if using functions of the form $f_k(X_{ig}) = \theta_{kd}$, we can introduce a global intercept term θ_0, which acts as a base value, and assign $\theta_{k1} = 0$, for $k = 1, 2, \ldots, K$. Again, the total degrees of freedom will be reduced by $K - 1$.

For categorical predictors, each f_k, and therefore $\sum_k f_k$, can be represented by a linear model. An all-inclusive approach is to let $K = N$ and write the underlying relationship as $f(\boldsymbol{X}) = \boldsymbol{J\Theta}$, where $\boldsymbol{J} = [\mathbf{1}\ \boldsymbol{J}_1\ \boldsymbol{J}_2 \cdots \boldsymbol{J}_N]$ and $\boldsymbol{\Theta} = [\theta_0, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N]$. In this model, $\boldsymbol{\theta}_g$ represents the coefficients specific to predictor g. In most cases, the degrees of freedom of this function will far exceed n. Therefore, it becomes necessary to encourage most $\boldsymbol{\theta}_g$ to have either zero (or negligible) magnitude, indicating that predictor g does not (significantly) contribute.

In the frequentist set-up, there are many flavours of penalty term which will have the desired effect. Perhaps the simplest of these is "variable subset selection", which enforces a penalty based only on the number of non-zero regression coefficients. An example penalty term is the Akaike Information Criterion (AKAIKE, 1974) which simply increases the residual

sum of squares by an amount proportional to the number of non-zero coefficients. Variable subset selection generally uses a stepwise search of the model space. Each model dictates which regression coefficients are non-zero, conditional on which the best fit can be calculated using the least squares estimates.

"Ridge regression" is a description given to methods which penalise based on the sum of the squares of regression coefficients (e.g. Zhang and Xu, 2005; Park and Hastie, 2008). By contrast, the LASSO method penalises according to the sum of their absolute values (Tibshirani, 1996). Generally, the penalty term is prefaced by a scale factor λ, so that as $\lambda \to 0$ the solution approaches the least squares estimates. Ridge regression and the LASSO can be compared by considering their effect on the best fit as λ is increased from zero. For ridge regression, the least squares estimates of the regression coefficients are reduced in a continuous fashion, only reaching zero when $\lambda = \infty$. For the LASSO, the estimates reach zero at different points, depending on the predictors' relative contributions to $f(\boldsymbol{X})$. This highlights the differences between each method's sparsity assumption. The former supposes that there are a few strong associations, while most predictors contribute only slightly; the latter supposes most predictors contribute in no way at all.

Many frequentist methods have Bayesian analogies. For example, variable subset selection equates to placing a point mass on elements of $\boldsymbol{\Theta}$ (e.g. Kuo and Mallick, 1998), ridge regression corresponds to a normal prior (e.g Zhang *et al.*, 2005; Wang *et al.*, 2005), while the LASSO relates to a double-exponential distribution (e.g. Yi and Xu, 2008; Hoggart *et al.*, 2008).

The use of mixture priors allows more complicated methods to be devised. Shotgun Stochastic Search (*SSS*; Hans *et al.*, 2007) is one of these. Given a prior probability of association $p \in (0, 1)$, it assigns the regression coefficient corresponding to the gth predictor the spike and slab prior distribution

$$\mathbb{P}(\boldsymbol{\theta}_g) = (1 - p)\delta_{\{0\}} + p\,\mathbb{N}(0, \sigma^2),$$

where $\delta_{\{0\}}$ represents a point mass function at 0 which "integrates" to 1, and $\mathbb{N}(0, \sigma^2)$ denotes a normal distribution with mean 0 and variance σ^2. *SSS* searches the model space in a stepwise fashion, at each step deciding whether to add in, swap out or remove a contributing predictor. The method calculates the posterior scores for all models within the "neighbourhood" of the current state; those models reachable by a single move of the type add, swap or remove. Based upon these scores, *SSS* constructs a proposal distribution from which it picks which model to move to next.

SSS keeps track of the top scoring models it explores, from which it estimates posterior probabilities of association for each predictor. The accuracy of these estimates depends on the extent that the model search succeeds in identifying the best models. In essence, *SSS* tries to approximate the complete space of models by its list of top scoring models, so the greater the proportion of posterior weight contained within this list, the more accurate the approximation will be. As Hans *et al.* discuss, rather than at each step automatically accepting the proposed move, they could instead adopt a conventional MCMC strategy and calculate an acceptance probability. The method could then calculate posterior estimates in the normal fashion, based on how often each predictor is included in the Markov Chain. The authors conclude, however, that their search is preferable.

For quantitative predictors, this category of underlying relationship takes the form of the generalized additive model (Hastie and Tibshirani, 1990), with a Bayesian version discussed in Ravikumar (2009). As with functional data analysis, these methods are suited for very small numbers of predictors and when prediction, rather than variable selection, is the main focus.

1.4.5 More Than One Group, Maximum Group Size Greater Than One

$$f(\boldsymbol{X}) = f_1(X_{G_{11}}, \ldots, X_{G_{1s_1}}) + f_2(X_{G_{21}}, \ldots, X_{G_{2s_2}}) + \cdots + f_K(X_{G_{K1}}, \ldots, X_{G_{Ks_K}})$$

Allowing both interactions and multiple groups of predictors to contribute to the underlying relationship has the potential of most accurately describing the true model. However, both decisions increase the size of the model space and so the difficulty of identifying this true model. Relatively speaking, a limited number of methods fall into this category.

Logic Regression (*Logic*) is one such method, suitable when the predictors are binary. *Logic* creates new predictors, each of which are logical functions of the original ones. For example, one new predictor might be $X_1^C \vee (X_2 \wedge X_3)$, where $\vee$ and $\wedge$ represent the Boolean functions "OR" and "AND", and X_1^C is the complement of X_1. This predictor takes value one if either X_1 is zero or both X_2 and X_3 are one. *Logic* then fits a linear model with these new predictors. Each predictor is allowed to feature in more than one group, which allows the search of a broader range of interactions. For example, while each of $f_1(X_1, X_2) = \theta_{10} + \theta_{11} X_1 \wedge X_2$ and $f_2(X_1, X_2) = \theta_{20} + \theta_{21} X_1^C \wedge X_2^C$ have 2 degrees of freedom, a linear combination of f_1 and f_2 remains a function of X_1 and X_2, but has degrees of freedom 3.

Logic operates in two flavours: it either explores the model space in a frequentist manner, seeking the best scoring set of new predictors (Ruczinski *et al.*, 2003); or it adopts a

Bayesian search, using MCMC to produce estimates of posterior probabilities of association (Kooperberg and Ruczinski, 2005). If the predictors are tertiary, the method suggests recoding each as two variables, whereby 0, 1 and 2 are transformed to $(0,0)$, $(1,0)$ and $(1,1)$, respectively. In genetic terms, the two new predictors represent the dominant and recessive components of the original variant. To apply *Logic* to quantitative predictors, it would be necessary to recode each predictor as one or more binary variables, for example, thresholding values in a manner similar to *CART*.

Multivariate Adaptive Regression Splines (Friedman, 1991) is a second method in this category, primarily designed for continuous predictors. *MARS* also places restrictions on the types of functions permitted, considering only products of hinge functions:

$$f_k = \Theta_k \prod_j h(G_{gj}, i_{G_{gj}}),$$

where either $h(g,i) = \max(0, X_g - X_{ig})$ or $h(g,i) = \max(0, X_{ig} - X_g)$. Each $h(g,i)$ is only non-negative on one side of its corresponding knot X_{ig}. Therefore, the product of functions of this type will be non-negative over an ever-decreasing proportion of the input space. As a result, *MARS* is able to model changes over very fine scales, allowing it to pick out local variation. Denison and Holmes (2003) consider a Bayesian version of this method.

Sparse Partitioning, the method to which I devote the remainder of this thesis, falls into this category, but unlike *Logic* and *MARS* it attempts to apply no restrictions to the functions.

1.4.6 Other Methods

So far, I have focused on regression methods which can be applied when the response is continuous. Partly, this is because a continuous response should provide more information than a binary one, a property which becomes increasingly important when considering interactions. Furthermore, any regression method suitable for continuous values can be either adapted for, or applied directly to, a binary response, whereas the converse is not true. Here, I mention methods for analysing binary response data, as well as one further method suitable for a continuous response. Most of these methods loosely fall into the category "one group, maximum group size greater than one".

Sparse combinatorial inference (Mukherjee *et al.*, 2009) considers the contingency table formed by a single group of interacting predictors ($K = 1$; $s_1 = 1, 2, 3, \ldots$). For example, for two binary predictors, the table will have four cells, counting the number of occurrences of $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$. To each cell, the method assigns a parameter, which represents the probability of samples corresponding to that cell having response value 1. On this basis,

each contingency table is scored in a Bayesian fashion, according to how well it fits the data. The method seeks to deduce the most plausible grouping using MCMC.

The approach of multifactor dimensionality reduction (Hahn *et al.*, 2002) is very similar, but set within a frequentist framework. First, the method splits the samples into a training and test set. For a given contingency table, it assigns each cell value either 1 ("high risk") or 0 ("low risk") according to the numbers of cases ($Y_i = 1$) and controls ($Y_i = 0$) in the training set to which this cell corresponds. The table is then scored by using these assignments for the test dataset, counting how many of the response values it correctly predicts. Rather than explore the model space in a stepwise fashion, multifactor dimensionality reduction exhaustively scores each possible contingency table, returning the best one found. As a table's prediction accuracy will depend on the choice of training and test sets, to obtain a reliable score it is necessary to repeat this procedure for a number of divides. The exhaustive nature of the search places a limit on the sizes of models and dataset the method can consider.

Verzilli *et al.* (2006) construct a Bayesian graphical model, one which considers the joint likelihood of the response and the predictors. Designed for association study data, the method searches for the best division of predictors into cliques, where each clique indicates dependency between the variants it contains. Each graph defines three types of predictor. Those "directly" associated with the phenotype lie in a clique containing the response. Those "indirectly" associated can be linked to the response via one or more other cliques. However, the majority of predictors occur in cliques completely disjoint from the response, indicating they are in no way associated. The graphical structure enables the method to account for LD, so hopefully allowing it to more accurately detect associations when strong correlations exist between predictors.

Mailund *et al.* (2006) also take a graph-based approach, devising a method which incorporates coalescent theory. For each locus, first they create the phylogenetic tree which explains that predictor and as many neighbours as possible. This tree will divide the samples according to the end branch on which each lies, so can be scored by comparing this partitioning with the response. By regressing on trees, rather than individual predictors, the method is able to consider LD and possible interactions over the (local) area on which the tree is defined.

BAMSE (Bayesian Association for Multiple SNP Effects; Albrechtsen *et al.*, 2007) is predominantly designed for a continuous response. The method considers multiple groups of associated predictors, each of which defines a "risk set" of samples and is assigned a mean phenotypic value. For example, one risk set might contain all samples with $X_1 > 1$ and $X_2 < 2$, while a second might include all samples with $X_3 > 0$. Although *BAMSE* allows multiple groups of associations, under my terminology it falls into the category "one group, maximum group size greater than one"; when a sample satisfies the conditions for two or more risk sets,

it is assigned only to the one with the highest phenotypic mean. Therefore, just like *CART*, each model dictates a partitioning of the samples, and can be described by a single design matrix $\boldsymbol{J}_1$, with corresponding parameter vector $\boldsymbol{\theta}_1$. The space of possible configurations of risk sets is explored using MCMC.

Two final methods are Combinatorial Partitioning (Nelson *et al.*, 2001) and BEAM (Zhang and Liu, 2007), which I discuss during the description of *Sparse Partitioning*.

Chapter 2

Sparse Partitioning

This chapter outlines the core of Sparse Partitioning*'s methodology, saving superfluous details for later.* Sparse Partitioning *is suitable only for problems with tertiary predictors, those that can be represented by values from 0, 1 or 2. These predictors are treated as categorical, so their order and the choice of labelling is irrelevant.*

2.1 Motivation

In the previous chapter, I expressed the underlying relationship as the sum of functions of groups of predictors:

$$f(\boldsymbol{X}) = f_1(X_{G_{11}}, \ldots, X_{G_{1s_1}}) + f_2(X_{G_{21}}, \ldots, X_{G_{2s_1}}) + \cdots + f_K(X_{G_{K1}}, \ldots, X_{G_{Ks_K}}),$$

with predictors free to feature in more than one group. Let's suppose we are given the groups of predictors and wish to explore possible sets of functions $\boldsymbol{f} = \{f_1, f_2, \ldots, f_K\}$. How many different forms are there for each f_k, bearing in mind we are considering categorical predictors? Let's examine the simplest case, a function of two binary predictors $f_k(X_g, X_{g'})$. In total, there are up to four distinct values (nodes) for $(X_g, X_{g'})$, namely, $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$. Each suitable function provides a mapping of each node to a real value: $f_k \colon \{0,1\}^2 \mapsto \mathbb{R}$.

The function need not permit different nodes to map to different values. Instead, it may insist, say, that $f_k(0,0) = f_k(0,1)$. The degrees of freedom of the function is equal to the number of free parameters. This will equal 4 if all nodes are allowed to map to different values, less than 4 if there are restrictions. Figure 2.1 displays the different functional forms possible for degrees of freedom 2, 3 and 4 (the case when the degrees of freedom equals 1 is ignored, as then the function would be trivial). In total, there are 14 possible forms, however, two of the forms with degrees of freedom 2 are disallowed: when $f(0,0) = f(0,1) \neq f(1,0) = f(1,1)$, the second predictor is redundant; similarly, this is the

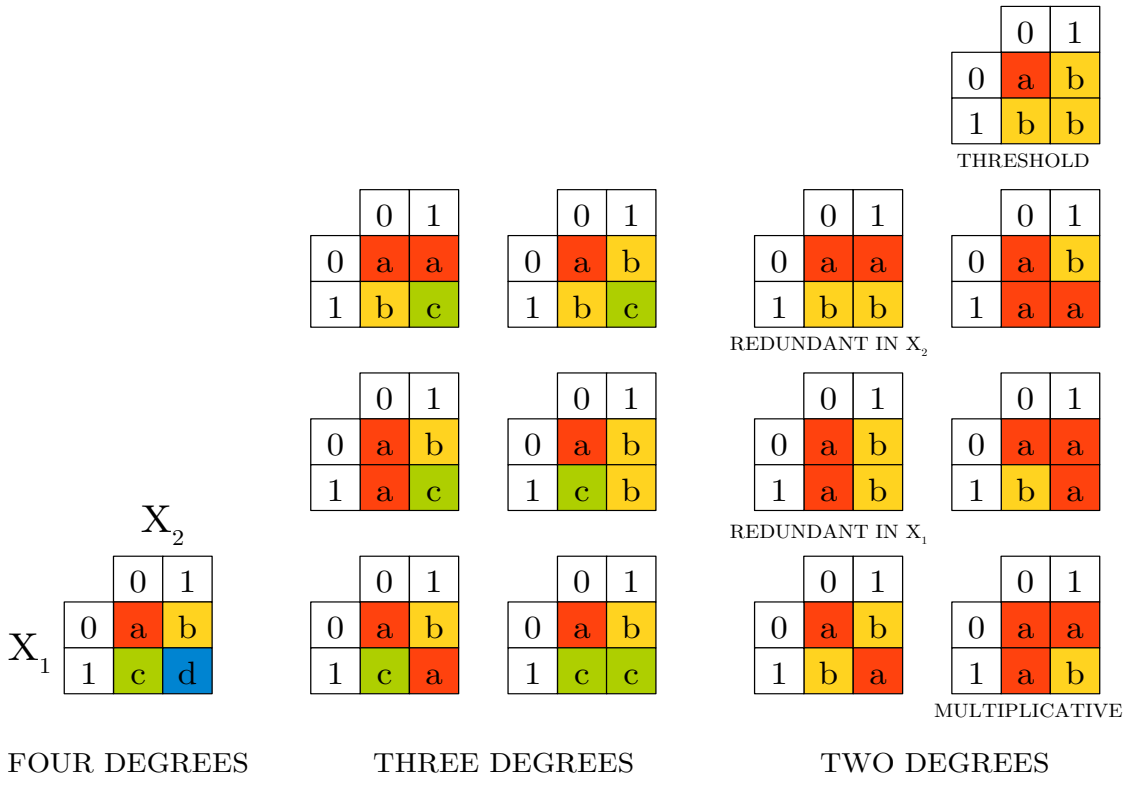


Figure 2.1: *Possible functional forms acting on a pair of binary predictors. Each grid demonstrates a possible form for $f(X_1, X_2)$. Within each grid, the colours indicate which nodes are mapped to the same value, so the total number of colours in a grid represents the degrees of freedom. Two grids represent redundant forms, the case when either X_1 or X_2 has no influence on the function. In addition, the standard "multiplicative" and "threshold" interaction models are highlighted.*

case for the first predictor if $f(0,0) = f(1,0) \neq f(0,1) = f(1,1)$.

Nonetheless, there remains a total of 12 possible functional forms. Some of these have obvious interpretations in genetics. For example, when $f(0,0) = f(0,1) = f(1,0) \neq f(1,1)$, this represents a multiplicative interaction, so that the effect of two variants is noticed only when both are mutant. By contrast, $f(0,0) \neq f(0,1) = f(1,0) = f(1,1)$ represents a threshold interaction, so only one variant need be mutated for its effect to become apparent.

As the diagram demonstrates, there are two steps in choosing a function of categorical predictors: deciding its degrees of freedom, then deciding its form. Ideally, a regression method would try to determine the correct degrees of freedom. If the method tries to fit a function with too few degrees of freedom, the fit will be inaccurate, as there will be nodes incorrectly assigned to equal values. If the method tries to fit a function with too many degrees of freedom, the method risks overfitting; if two nodes are assigned to different values, when in fact their underlying values are the same, the method will be fitting the noise present in the data. Additionally, most methods employ a trade-off between the fit of the model and a penalty

based on the complexity of the function. Therefore, fitting a function with excessive degrees of freedom will lead to unnecessary penalisation.

So, even for simple underlying relationships containing groups of two binary predictors, it is easy to appreciate how much harder testing nonlinear models is compared to testing linear models. For a linear method, when only one predictor is allowed in each group, the form of each function is automatic; f_k assigns each of the two nodes to different values. This remains the case regardless of the number of groups. By contrast, a nonlinear method featuring one group of two predictors has 12 different functions to consider. If instead there are two groups of size two, there will be 144 possibilities, and so on. As the number of nodes in each group grows, either by increasing the number of categories of predictors or the number of predictors involved, the difference between the complexities of the linear and nonlinear methods grows further.

Without doubt, it is necessary to limit the number of possible functional forms. Logic Regression's approach is to consider only Boolean functions, therefore insisting f_k has 2 degrees of freedom. If a group of predictors has d nodes, this reduces the number of possible functions to at most $2^{d-1} - 1$[1]. The inherent assumption of *Logic* is either that we are certain that the true functions are Boolean or that, if this is a simplifying assumption, the computational advantages of this assumption outweigh the loss of accuracy.

Sparse Partitioning takes the opposite approach to *Logic*, insisting the degrees of freedom equals the number of nodes. In this case, as with linear functions, there is only one possible form for each function and therefore only one functional form per model. Undoubtedly, this approach will lead to overfitting, as it lacks the capacity to reduce the degrees of freedom. I believe that, as the number of nodes remains manageable for tertiary predictors, the damage of this overfitting will be compensated for by the increased accuracy provided. In essence, I feel that the dangers of overfitting are less than the dangers of underfitting.

Comparison with Combinatorial Partitioning

My approach starts out similar to that of Combinatorial Partitioning (Nelson *et al.*, 2001), which also looks at the many different forms a function can take. But whereas I settled upon a single form for each function, Nelson *et al.* opt for an exhaustive search, examining all possible functional forms for all possible degrees of freedom. Their paper demonstrates the limitations necessary to make such a search feasible. They restrict to a single group ($K = 1$)

[1] Consider a Boolean mapping of d nodes. Each node can be mapped to either 1 or 0, so there are 2^d possible configurations. Half of these are redundant (toggling the image of each node will not affect the function), while one case is trivial. There might be additional redundancies within these $2^{d-1} - 1$ possibilities, as there will be cases when the values of one or more predictors are of no consequence.

of at most pairwise interactions ($s_1 = 2$). For tertiary predictors, there are 21,147-1 possible functional forms, ranging from 2 to 9 degrees of freedom. Therefore, the total run time takes approximately 20,000 times as long as *Pairs*. While such an approach is plausible for small predictor sets (they demonstrate for $N = 18$), it is certainly not suitable for high-dimensional problems, nor if we wish to explore higher values of K and/or s_k.

2.2 Partitioning Notation

The fundamental premise of *Sparse Partitioning* is that the search for associated predictors corresponds to a search for high scoring partitions. Consider how the underlying relationship groups predictors:

$$\begin{aligned} f(\boldsymbol{X}) &= f_1(X_{G_{11}}, \ldots, X_{G_{1s_1}}) + f_2(X_{G_{21}}, \ldots, X_{G_{2s_2}}) + \cdots + f_K(X_{G_{K1}}, \ldots, X_{G_{Ks_K}}) \\ &= f_1(X_{\boldsymbol{G}_1}) + f_2(X_{\boldsymbol{G}_2}) + \cdots + f_K(X_{\boldsymbol{G}_K}). \end{aligned}$$

The sets $\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_K$ index groups of associated predictors. For the moment, suppose they are disjoint, so each predictor appears in at most one set. If we let $\boldsymbol{G}_0$ represent the "null group" — the group of predictors in no way associated with the response — then $\mathbb{G} = \{\boldsymbol{G}_0, \boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_K\}$ defines a partitioning of {1,2,...,N}. Neither the labelling of groups, nor the ordering of predictors within groups, is important.

It is conceivable that some predictors might feature in more than one group of associations. To allow this, I expand the predictor set to contain C copies of each predictor and increase the total number of predictors, N, accordingly. I explain the reason I opt for this approach, rather than simply relaxing the condition of disjointness, when discussing the prior probabilities assigned to different partitions. In general, I describe *Sparse Partitioning*'s method supposing $C = 1$, then explain the changes brought about when C is increased.

A partition can also be described by the vector $\boldsymbol{I} = (I_1, I_2, \ldots, I_N)$, where I_g indicates to which group predictor g belongs. Although this notation is very inefficient when only a small proportion of predictors are associated, it proves useful later on and emphasises that the labelling within groups is irrelevant. As there is a one-to-one relationship between partitions $\mathbb{G}$ and indicator sets $\boldsymbol{I}$, I will use these two terms interchangeably. Finally, let the set $S_{\boldsymbol{I}} \subseteq \{1, 2, \ldots, N\}$ index the predictors partition $\boldsymbol{I}$ declares associated: $g \notin \boldsymbol{G}_0 \Leftrightarrow I_g \neq 0 \Leftrightarrow g \in S_{\boldsymbol{I}}$.

Here, I give an example of a partitioning and the underlying relationship to which it refers:

$$f(\boldsymbol{X}) = f_1(X_1, X_2) + f_2(X_5) \;\Leftrightarrow\; \boldsymbol{I} = (\overbrace{1\ \ 1}^{\boldsymbol{G}_1}\overbrace{0\ \ 0}^{\boldsymbol{G}_0}\overbrace{2}^{\boldsymbol{G}_2}) \;\Rightarrow S_{\boldsymbol{I}} = \{1, 2, 5\}.$$

This simple example contains five predictors, of which three contribute towards the true underlying relationship: $f(\boldsymbol{X}) = f_1(X_1, X_2) + f_2(X_5)$. For this underlying relationship, there are two non-empty non-null groups: one labelled G_1, containing the predictors X_1 and X_2; the other labelled G_2, containing X_5. The remaining predictors, X_3 and X_4, form the null group. This partitioning corresponds to the indicator vector $\boldsymbol{I} = (1, 1, 0, 0, 2)$. When wishing to represent a partition graphically, I will list the non-null groups. Therefore, I would write this partition as $\{1, 2\}$ $\{5\}$.

Introducing the partitioning concept allows us to deconstruct each underlying relationship into a partition $\mathbb{G}$ and a corresponding set of functions $\boldsymbol{f} = \{f_1, f_2, \ldots, f_K\}$. A regression method which explores the space of underlying relationships would consider models of the form $\{\mathbb{G}, \boldsymbol{f}\}$. In my opinion, the information contained within $\mathbb{G}$ is far more important than that provided by $\boldsymbol{f}$. If we were able to determine the true partition, it would be relatively straightforward to interrogate $\boldsymbol{f}$ in a follow-up experiment. As such an experiment would be matter-of-course in validating any results, this would provide no extra work.

Therefore, I made the decision that my method would only be concerned with exploring the space of partitions and would treat $\boldsymbol{f}$ as a nuisance parameter. This approach greatly reduces the complexity of the model search.

Secondly, my feeling is that "model averaging" methods, those which draw conclusions from a number of plausible models, perform better than "mode seeking" methods, which make inferences from a single best scoring model. Especially when the number of predictors far exceeds the number of samples, it seems unrealistic to hope to correctly deduce the exact true underlying relationship. If one partition scores much higher that all others, then both a mode seeking and a model averaging approach have the potential to identify this partition. If, however, there are a number of high scoring partitions, a method which returns only the best will not appreciate models which perform almost as well. Additionally, a model averaging approach will allow for the overlap between different models; when assessing the evidence that a predictor is associated, it will consider the different ways this predictor might be associated.

As a Bayesian method, *Sparse Partitioning* concentrates on evaluating the posterior distribution of the parameters. Ideally, it would be possible to calculate this distribution explicitly, as then we could extract the answers to any questions we pleased. As this is wholly infeasible for any reasonably sized problem, *Sparse Partitioning* focuses on two more specific questions: firstly, "Which predictors contribute to the underlying relationship?" and secondly, "Which predictors interact?" In terms of the partitioning notation, these questions correspond to asking which predictors are not in the null group and which predictors are in the same non-null group. The former will often tie in nicely with our prior information, which will likely consider

the probabilities that different predictors are associated.

2.3 Bayesian Framework

All Bayesian methods consist of three stages: having identified the unknown parameters, they must design a prior distribution, formulate a likelihood, then attempt to calculate the posterior. For the moment, let's suppose the unknown parameters are simply $\mathbb{G}$ and $\boldsymbol{f}$, while $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent the data. Later on, we will encounter other unknown parameters and the form of the data will change, however, I will deal with these considerations when they arise.

Before launching into *Sparse Partitioning*'s methodology, it is first necessary to deal with a slight technicality. Suppose we have a "designed experiment", one in which we can dictate the observed values of the predictors. The likelihood of the data will be

$$\begin{aligned}\mathbb{P}(\boldsymbol{X}, \boldsymbol{Y}|\mathbb{G}, \boldsymbol{f}) &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{f}) \times \mathbb{P}(\boldsymbol{X}|\mathbb{G}, \boldsymbol{f}) \\ &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{f}).\end{aligned}$$

The last term has vanished because the observed values of $\boldsymbol{X}$ are selected in advance, so their likelihood is 1.

In many cases, we are not at liberty to decide the observed predictor values. For example, in an association study we typically pick samples based on their phenotype, then type these samples to identify their variants. There are occasions when we might select samples in the reverse manner. If we are testing the effect of a particular variant, we might desire equal numbers of samples with and without the mutation, so first perform a low-cost screening to find two such groups. We might be able to perform such a selection for two, possibly three variants, but certainly not for a reasonable number.

For this reason, a fully Bayesian method should also consider the likelihood of the predictor values. Let's express this as $\mathbb{P}(\boldsymbol{X}|\epsilon)$, where ϵ is a parameter vector separate from $\mathbb{G}$ and $\boldsymbol{f}$. Fortunately, as explained in Gelman *et al.* (2004), it is reasonable to ignore ϵ's presence. The posterior distribution can be written as

$$\begin{aligned}\mathbb{P}(\mathbb{G}, \boldsymbol{f}, \epsilon|\boldsymbol{X}, \boldsymbol{Y}) &\propto \mathbb{P}(\boldsymbol{X}, \boldsymbol{Y}|\mathbb{G}, \boldsymbol{f}, \epsilon) \times \mathbb{P}(\mathbb{G}, \boldsymbol{f}, \epsilon) \\ &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{f}, \epsilon) \times \mathbb{P}(\boldsymbol{X}|\mathbb{G}, \boldsymbol{f}, \epsilon) \times \mathbb{P}(\mathbb{G}, \boldsymbol{f}, \epsilon) \\ &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{f}) \times \mathbb{P}(\boldsymbol{X}|\epsilon) \times \mathbb{P}(\mathbb{G}, \boldsymbol{f}, \epsilon).\end{aligned}$$

The last line follows because the likelihood of $\boldsymbol{Y}$, as governed by the regression equation, does not involve ϵ, while the likelihood of $\boldsymbol{X}$ does not depend on $\mathbb{G}$ nor $\boldsymbol{f}$. It seems reasonable

to suppose the prior for ϵ is independent of all other unknown parameters: $\mathbb{P}(\mathbb{G}, \boldsymbol{f}, \epsilon) = \mathbb{P}(\mathbb{G}, \boldsymbol{f}) \times \mathbb{P}(\epsilon)$. Therefore, the posterior distribution can be broken into two parts, which have in common only the observed values for $\boldsymbol{X}$ and $\boldsymbol{Y}$:

$$\mathbb{P}(\mathbb{G}, \boldsymbol{f}, \epsilon | \boldsymbol{X}, \boldsymbol{Y}) \propto \mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}, \mathbb{G}, \boldsymbol{f}) \times \mathbb{P}(\mathbb{G}, \boldsymbol{f}) \ . \ \mathbb{P}(\boldsymbol{X} | \epsilon) \times \mathbb{P}(\epsilon).$$

If we are only interested in the marginal posterior distribution $\mathbb{P}(\mathbb{G}, \boldsymbol{f} | \boldsymbol{X}, \boldsymbol{Y})$, we can integrate with respect to ϵ and obtain

$$\mathbb{P}(\mathbb{G}, \boldsymbol{f} | \boldsymbol{X}, \boldsymbol{Y}) \propto \mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}, \mathbb{G}, \boldsymbol{f}) \times \mathbb{P}(\mathbb{G}, \boldsymbol{f}),$$

demonstrating why it is justifiable to ignore ϵ.

2.4 Prior Distribution

The joint prior $\mathbb{P}(\mathbb{G}, \boldsymbol{f})$ can be written as $\mathbb{P}(\mathbb{G}) \times \mathbb{P}(\boldsymbol{f} | \mathbb{G})$, so I consider the two distributions separately. The dependency of $\boldsymbol{f}$ on $\mathbb{G}$ exists because the partition determines the structure and degrees of freedom of each function. However, this dependency proves very slight and, as I will show later, can be removed without consequence.

The prior distribution should be flexible, to allow the user to incorporate their own beliefs. Very often, these beliefs will focus on how likely it is that each predictor is associated. Let p_g represent the user's prior probability that predictor g is associated and let $\boldsymbol{p} = \{p_1, p_2, \ldots, p_N\}$. As discussed in the introduction, these probabilities will typically reflect a belief in sparsity, which supposes that the expected number of causal predictors is small. Here, I explain further why an assumption of this nature is necessary.

The number of samples required to detect an underlying relationship is linked to its complexity. Suppose that $f(\boldsymbol{X})$ is limited to a single function ($K = 1$). To correctly identify a function with D degrees of freedom, there must be at least D distinct vector values observed for the associated predictors. In terms of Figure 2.1, this corresponds to there being at least one sample within each group of cells of a different colour. The resolution of the method will then depend on how many samples correspond to each node. As the number of predictors contributing towards the underlying relationship increases, so will the number of distinct samples required. Therefore, for high-dimensional problems, the sparsity assumption becomes crucial.

I have come across many methods that consider the maximum number of associations it is realistic to detect in order to determine an upper bound for the *a priori* expected number of associations. For example, Zhang *et al.* (2005) argue that there should be at most $\sqrt{n}$

associations (which in their model corresponds to $2\sqrt{n}$ degrees of freedom), Zhang and Liu (2007) discuss values no higher than $\log_3(N/2)$, while Mukherjee *et al.* (2009) reason that there should be on average 15 samples corresponding to each degree of freedom. If the user is uncomfortable with choosing p_g according to the limitations of the method, the prior mean can be reinterpreted as a belief concerning the number of "strong associations", and the method can be viewed instead as a search for strong associations.

2.4.1 Partition Prior, $\mathbb{P}(\mathbb{G})$

The construction of $\mathbb{P}(\mathbb{G})$, the prior for the partition, is based on the vector of marginal prior probabilities of association, $\boldsymbol{p}$. Bearing in mind that each partition $\mathbb{G}$ corresponds to an indicator vector $\boldsymbol{I}$, we wish to construct $\mathbb{P}(\boldsymbol{I})$ so that it has the required marginal probabilities of association:

$$\mathbb{P}(I_g \neq 0) = \sum_{\boldsymbol{I}:I_g\neq 0} \mathbb{P}(\boldsymbol{I}) = p_g.$$

For a given partition $\boldsymbol{I}$, let the equivalence class $[\boldsymbol{I}]$ contain all partitions that declare the same predictors associated: $\boldsymbol{I'} \in [\boldsymbol{I}] \Leftrightarrow S_{\boldsymbol{I'}} = S_{\boldsymbol{I}}$. I define the "probability of an equivalence class" $\mathbb{P}([\boldsymbol{I}])$ as the sum of the probabilities of partitions within it:

$$\mathbb{P}([\boldsymbol{I}]) := \sum_{\boldsymbol{I'}\in[\boldsymbol{I}]} \mathbb{P}(\boldsymbol{I'}).$$

We can achieve the desired probabilities of association by insisting

$$\mathbb{P}([\boldsymbol{I}]) = \prod_{j\in S_{\boldsymbol{I}}} p_j \prod_{j\notin S_{\boldsymbol{I}}} (1-p_j),$$

as then the marginal probability that predictor g is associated will be

$$\begin{aligned}\mathbb{P}(I_g \neq 0) = \sum_{\boldsymbol{I}:I_g\neq 0} \mathbb{P}(\boldsymbol{I}) = \sum_{[\boldsymbol{I}]:g\in S_{\boldsymbol{I}}} \mathbb{P}([\boldsymbol{I}]) &= \sum_{[\boldsymbol{I}]:g\in S_{\boldsymbol{I}}} \left(\prod_{j\in S_{\boldsymbol{I}}} p_j \prod_{j\notin S_{\boldsymbol{I}}} (1-p_j) \right) \\ &= p_g \sum_{[\boldsymbol{I}]:g\in S_{\boldsymbol{I}}} \left(\prod_{g\neq j\in S_{\boldsymbol{I}}} p_j \prod_{g\neq j\notin S_{\boldsymbol{I}}} (1-p_j) \right).\end{aligned}$$

If all possible equivalence classes are achievable — by which I mean that given any subset of $\{1, 2, \ldots, N\}$, the underlying relationship permits at least one partition $\boldsymbol{I}$ whose set of associations $S_{\boldsymbol{I}}$ matches this subset — then

$$\sum_{[\boldsymbol{I}]:g\in S_{\boldsymbol{I}}} \left(\prod_{g\neq j\in S_{\boldsymbol{I}}} p_j \prod_{g\neq j\notin S_{\boldsymbol{I}}} (1-p_j) \right) = \prod_{j\neq g} (p_j + (1-p_j)) = 1.$$

Therefore, $\mathbb{P}(I_g \neq 0) = p_g$, as desired. Later on, I discuss the approximation used when some equivalence classes are not achievable.

Weighting of Partitions within Equivalence Classes

Having determined $\mathbb{P}([\boldsymbol{I}])$, it remains to decide how to divide this probability across the partitions within the equivalence class. If we insist that we are able to precompute $\mathbb{P}(\boldsymbol{I})/\mathbb{P}([\boldsymbol{I}])$, then this value must be independent of the elements contained within $S_{\boldsymbol{I}}$. Were the weighting to depend on $S_{\boldsymbol{I}}$, then it would be necessary to examine every partition and equivalence class individually. This task would be of similar order of magnitude to the task of exhaustively evaluating the posterior distribution, which is not possible for reasonably sized datasets.

If we do not require that $\mathbb{P}(\boldsymbol{I})/\mathbb{P}([\boldsymbol{I}])$ be precomputed, we could calculate its value on-the-fly. However, I feel that doing so would greatly increase the computation time. I can think of no efficient means of storing weightings for each equivalence class, so for classes visited multiple times, these values would have to be calculated repeatedly. In any event, this becomes a moot point, as I decided to assign equal weighting to members of $[\boldsymbol{I}]$. Shortly, I discuss the rationale for this decision, but before that, I explain its implementation.

To weight members equally, it is necessary to calculate the size of each equivalence class. The size of $[\boldsymbol{I}]$ depends only on the number of predictors declared associated by each of its members. Let s denote the number of associations: $|S_{\boldsymbol{I}}| = s$. The size of $[\boldsymbol{I}]$ equals the number of distinct partitions of s elements. Unrestricted, this equals $B(s)$, the sth Bell number. However, *Sparse Partitioning* limits K and S, the maximum number of groups and the maximum number of elements within each group, so it is necessary to calculate the "truncated Bell numbers" $B(s, K, S)$. These can be worked out in a recursive fashion, similar to how one might calculate the original Bell numbers. Let a_j denote the number of groups of size j, for $j = 1, 2, \ldots, S$. Then

$$\begin{aligned} B(s, K, S \,|\, a_1, a_2, \ldots, a_{S-1}, a_S) = {} & B(s, K, S \,|\, a_1 - 1, a_2, \ldots, a_{S-1}, a_S) + \\ & B(s, K, S \,|\, a_1 + 1, a_2 - 1, \ldots, a_{S-1}, a_S)\,(a_1 + 1) + \cdots + \\ & B(s, K, S \,|\, a_1, a_2, \ldots, a_{S-1} + 1, a_S - 1)\,(a_{S-1} + 1), \end{aligned}$$

with boundary condition

$$B(0, K, S \,|\, a_1, a_2, \ldots, a_{S-1}, a_S) = \begin{cases} 1, & \text{if } 0 = a_1 = a_2 = \ldots = a_S, \\ 0, & \text{otherwise.} \end{cases}$$

To calculate all possible $B(s, K, S \,|\, a_1, a_2, \ldots, a_{S-1}, a_S)$ requires of the order $KS \times K^S$ recursions. This computation time is insignificant for reasonable K and S. The only caveat is that

many programming languages have a maximum integer limit, so before each recursion, it is necessary to check whether this might be exceeded and round off values when necessary.

As I decided to weight partitions within an equivalence class equally, it is only necessary to calculate the partial sums

$$B(s, K, S) = \sum_{0 \leq a_1, a_2, \ldots, a_S \leq K} B(s, K, S \,|\, a_1, a_2, \ldots, a_{S-1}, a_S),$$

allowing us to set $\mathbb{P}(\boldsymbol{I})/\mathbb{P}([\boldsymbol{I}]) = B(s, K, S)^{-1}$. Written in full, the partition prior is

$$\mathbb{P}(\mathbb{G}) = \mathbb{P}(\boldsymbol{I}) = B(|S_{\boldsymbol{I}}|, K, S)^{-1} \prod_{g \in S_{\boldsymbol{I}}} p_g \prod_{g \notin S_{\boldsymbol{I}}} (1 - p_g).$$

Later on, for convenience, I will use $B(\mathbb{G})$ to denote $B(|S_{\boldsymbol{I}}|, K, S)$.

What other weightings could I have used? Whatever the choice, it would remain necessary to calculate all possible values of $B(s, K, S \,|\, a_1, a_2, \ldots, a_{S-1}, a_S)$, so a wealth of information would be available at no extra cost. From these values, it would be possible to reflect in the prior not only the probability that each predictor is declared associated, but also the belief in the number of groups of associations or the number of interactions. For example, for models declaring three associations, we could specify the prior probabilities that the true partition contains one, two or three non-null groups or, correspondingly, that the model features three, one or no interactions.

Given my choice to weight equally all partitions in the same equivalence class, the plots in the left column of Figure 2.2 show the effect this has on the the prior distribution for the number of groups of associations. Similarly, the right column shows how this decision affects the prior distribution for the number of interactions. For all plots, K and S are kept at their default values of 4, N is set to 10,000 and each predictor is assigned the same prior probability of association. In the top plots, each line corresponds to a different prior probability of association, with the expected number of associations ranging from 1 to 9. As the prior probability increases, so does the weighting assigned to higher numbers of groups/interactions.

When I experimented with different choices of weighting, it was easiest to manipulate the spread of probabilities given the number of predictors associated. The bottom plots look more closely at the case when $p_g = 5/N$. Each line corresponds to a particular number of associations, so once again it is clear that as this value increases, more emphasis is placed on higher numbers of groups/interactions. The dark green line corresponds to the case when three predictors are declared associated and shows how the uniform weighting places most

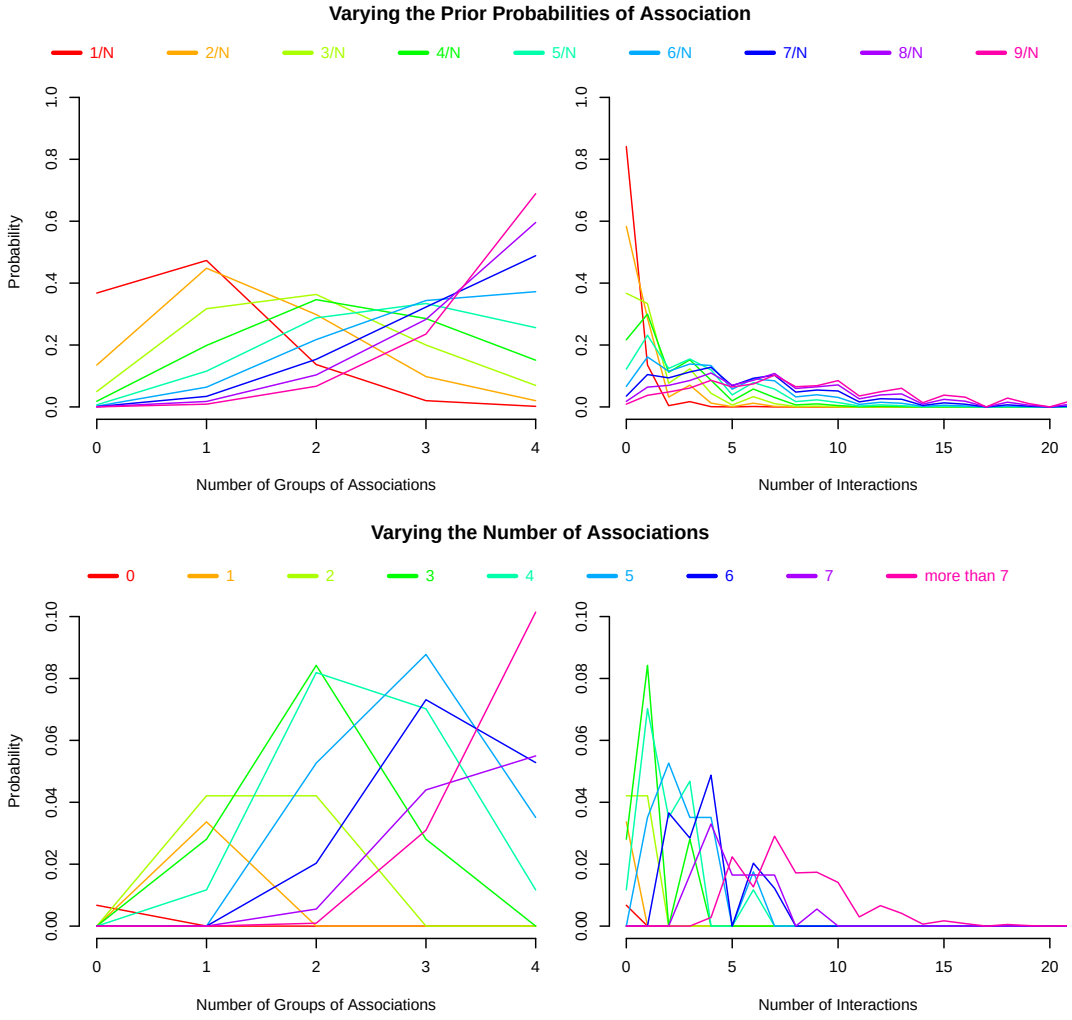


Figure 2.2: *The effect of a uniform prior weighting across equivalence classes. The left hand plots show how the decision to weight all partitions in an equivalence class equally determines the prior weight for the number of groups of associations. The right hand plots show how this decision effects a prior on the total number of interactions. In all cases, the maximum number of groups, K, and number of elements in each group, S, are set to 4, the total number of predictors, N, equals 10,000 and all predictors are assigned the same prior probability of association. In the top row, the different lines show the effect of varying the prior probability of association assigned to each predictor. As this probability is increased, the lines peak later as a larger share of the prior weighting is assigned to higher numbers of groups/interactions. The bottom row focuses on the case when $p_g = 5/10000$, each line corresponding to a different number of associations. For example, when there is one association (the orange line), there can only be one group and no interactions; whereas when there are more than 7 interactions (the pink line), there must be at least two groups and no fewer than four interactions.*

of the probability on there being two groups. If we considered this spread inappropriate, it would just be a matter of selecting new probabilities for there being one, two or three groups, effectively moving each point on the line up or down as desired.

It is interesting to note that a uniform weighting places a high prior probability $(1-B(\mathbb{G})^{-1}$ on the existence of at least one interaction, even though relatively few interactions have so far been found and verified. However, in my opinion, the lack of known interactions must to some extent be due to how hard they are to identify, coupled with how rarely they are searched for. It is for this reason that I am satisfied that a uniform weighting is a reasonable choice.

Prespecification of K and S

To enable precalculation of $B(s, K, S)$, and also to allow allocation of sufficient memory, *Sparse Partitioning* requires that the maximum number of groups, K, and maximum number of predictors in each group, S, are set in advance. If we wanted to consider all possible models, K and S should be set no smaller than N, to ensure the two most extreme underlying relationships are possible: either N groups of size one or one group of size N. However, in practice, such values would lead to unrealistic amounts of unnecessary computation and memory demands. The prior probability assigned to partition $\boldsymbol{I}$ decreases rapidly as the size of $S_{\boldsymbol{I}}$ increases. Therefore, partitions declaring many associations are incredibly unlikely to contribute to the posterior estimates and can safely be overlooked. As a result, I suggest K and S are set as small as possible without having a noticeable effect on the method's calculation of the posterior. In general, I have found that $K = 4$ and $S = 4$ are adequate, in that these values very rarely appear to hinder the search of the model space. Of course, should the user wish, they can restrict K and S further, if they wanted to exclude certain models entirely.

The calculation of $\mathbb{P}(I_g \neq 0)$ assumed that $K \times S \geq N$, as then the maximum number of predictors associated is greater than the size of the predictor set, so all possible equivalence classes are achievable. When this condition does not hold, the error involved can be calculated for the case that all prior probabilities of association are equal ($p_g = p$, for $g = 1, 2, \ldots, N$):

$$\begin{aligned}\mathbb{P}(I_g \neq 0) &= p \sum_{s=0}^{KS-1} \binom{N-1}{s} p^s (1-p)^{N-1-s} \Bigg/ \sum_{s=0}^{KS} \binom{N}{s} p^s (1-p)^{N-s} \\ &= p\ \mathbb{P}(s \leq KS-1 | s \sim \mathbb{B}(p, N-1))\ /\ \mathbb{P}(s \leq KS | s \sim \mathbb{B}(p, N)),\end{aligned}$$

where $\mathbb{B}(a, b)$ denotes a binomial distribution with a trials and probability of success b. Using a normal approximation for each binomially distributed variable, we obtain

$$\mathbb{P}(I_g \neq 0) = p\ \Phi\left(\frac{KS - \frac{1}{2} - (N-1)p}{\sqrt{p(1-p)(N-1)}}\right) \Bigg/ \Phi\left(\frac{KS + \frac{1}{2} - Np}{\sqrt{p(1-p)N}}\right),$$

where Φ is the cumulative density function for a standard normal. For small p, the value of $\mathbb{P}(I_g \neq 0)$ is affected most by the prior mean, Np. I suggested setting $K = 4$ and $S = 4$; entering these values into the equation above, we find that the actual prior probability of association used by *Sparse Partitioning* lies within 1% of the desired value p even when the prior mean is as high as 9.

Multiple Copies of Predictors

There are situations when we might wish to consider predictors featuring in multiple non-null groups. For example, in an association study, we might consider that a genetic variant affects a phenotype via more than one pathway. In order to allow such situations, but without disrupting the disjointness of groupings, I consider multiple copies of each predictor, the number of copies being determined by the parameter C. For example, if $C = 3$, the original predictor set is expanded to contain 3 copies of each predictor and so its size is increased to $3 \times N$.

To compensate for this change, the prior probability of association for each copy of predictor g is set to $1 - \sqrt[C]{1 - p_g}$. As the prior probabilities for each copy are independent, this ensures the correct marginal probability:

$$\begin{aligned}\mathbb{P}(\text{at least one copy of } X_g \text{ associated}) &= 1 - \mathbb{P}(\text{zero copies of } X_g \text{ associated}) \\ &= 1 - \left(\sqrt[C]{1 - p_g}\right)^C,\end{aligned}$$

which equals p_g as desired. Notice that this set-up effects a binomial distribution for the number of associated copies of each predictor. It is difficult to surmise whether a different breakdown would be more appropriate. However, if the user is adamant that an alternative choice is more suitable, they can set $C = 1$ and manually add copies of each predictor, specifying the individual probabilities they desire.

Allowing multiple copies of each predictor creates an element of duplication within the space of partitions. For example, a partition in which two copies of predictor g feature in the same non-null group effects the same form for the underlying relationship as the partition with one of these copies removed. The prior weighting for this underlying relationship will be increased by a factor of $1 + \mathbb{O}(p_g)$, but for small values of p_g this will be negligible. As with K and S, it is necessary to specify C in advance. I will explain that its value has minimal effect on computation time, but show in the simulation studies that larger values can be beneficial. Therefore, I recommend a generous setting, such as $C = 3$.

An alternative, and more obvious, solution would be to relax the condition of disjointness and consider groupings instead of partitions. However, after a few tries, I found this approach

produced too many complications, particularly when trying to devise a suitable prior. Suppose we considered each non-null group an independent sampling from $\{1, 2, \ldots, N\}$ and set to $1 - \sqrt[K]{1 - p_g}$ the prior probability that predictor g appeared in each. If duplicate models were not accounted for, there would be, for example, K^2 groupings which declared associated only predictors 1 and 2. Of these, K would feature an interaction, while $K(K-1)$ would be additive. It is undesirable that the relative prior weightings of these two possibilities depended on the number of non-null groups allowed, especially as this upper bound exists mainly for computational reasons, not due to prior belief. I believe it would be very difficult to assign prior weightings which offset the bias of K.

It was suggested to me by Terry Speed that "lattice graph" theory might provide a means for relaxing disjointness, while maintaining uniqueness, and allow construction of a suitable prior for groupings. Even so, I felt that approaches of this nature would introduce insurmountable challenges in the sampling steps used for posterior estimation.

Forced Inclusion

One of *Sparse Partitioning*'s strengths is its ability to accept individual prior probabilities of association for each predictor, rather than requiring them all to be the same. In particular, the user can insist a predictor is associated by setting the corresponding probability to 1. Generally, this approach indicates that the user is certain a predictor is associated, however, they might simply wish to investigate the consequences if this were the case.

Special care should be taken when C is greater than 1. If p_g is set to 1, the prior probability of association for each copy of predictor g will be $1 - \sqrt[C]{1 - 1} = 1$, indicating that all copies contribute to the response. This might not be what the user has in mind, as they might only be sure the first copy is associated. One solution is to manually append additional copies of each predictor to the matrix $\boldsymbol{X}$ and apply *Sparse Partitioning* with $C = 1$. The user would then be able to individually specify prior probabilities for each copy of each predictor, avoiding the danger of multiple copies of a single predictor being forcibly included. To avoid this hassle, *Sparse Partitioning* allows the user to enter a list of predictors to be always included in the current model. For these predictors, the corresponding elements of $\boldsymbol{p} = \{p_1, p_2, \ldots, p_N\}$ then specify the probabilities that additional copies are associated.

Fixed or Variable p_g

In *Sparse Partitioning*, the values of p_g are fixed for the duration of the method and, as a result, the partition prior remains constant. An alternative approach, adopted by Zhang *et al.* (2005) and Mukherjee *et al.* (2009), among others, is to introduce a hierarchical prior set-up where $\boldsymbol{p}$ is allowed to vary and provided with its own prior distribution. In this case,

a posterior estimate that predictor g is associated could be obtained directly by calculating the posterior mean of p_g, rather than by evaluating $\mathbb{P}(I_g \neq 0)$. However, this set-up would only become worthwhile if we possessed additional prior information about p_g; for example, if we had an idea of the overall spread of values for p_g or a sense of how these values varied relative to each other. In my opinion, it is unrealistic to think we have knowledge about the distribution of $\boldsymbol{p}$ above a belief in its mean, so I decided against this approach. Later on, however, I revisit this discussion, explaining the effect that such a change might have on the method.

2.4.2 Function Prior, $\mathbb{P}(\boldsymbol{f}|\mathbb{G})$

As *Sparse Partitioning* treats the predictors as categorical, each function will take the form $f_k(X_{\boldsymbol{G}_k}) = \theta_{kd}$, where d denotes which image corresponds to the node $X_{\boldsymbol{G}_k}$. If the nodes are labelled from 1 to d_k, this function can be fully described by the vector $\boldsymbol{\theta}_k = \{\theta_{k1}, \theta_{k2}, \ldots, \theta_{kd_k}\}$. When discussing the motivation for *Sparse Partitioning*, I explained my decision that each function should be as general as possible and therefore f_k have degrees of freedom equal to d_k, the number of nodes observed.

This decision already turns out to have a useful property. If we were to consider variable functional forms for each partition, we would have to decide the prior weightings for forms on an *ad-hoc* basis. For example, the number of functional forms involving two binary predictors will depend on whether 3 or 4 nodes have been observed. As *Sparse Partitioning* uses the most general function in each case, this is not an issue. For a method that intends to sample from the posterior distribution via MCMC sampling, a reduction in work load is always desirable, as this will increase the number of iterations possible.

When there are two or more groups, it is prudent to consider the issue of identifiability (although this is more of a concern for frequentist methods than Bayesian ones). Identifiability can be achieved by introducing a global intersect term θ_0 and then insisting for each non-empty non-null group a sample i_k is chosen and state $X_{i_k\boldsymbol{G}_k}$ mapped to zero. This corresponds to setting $\theta_{kd} = 0$ for one value of d. The choice of which node is mapped to zero has an effect on the posterior calculation, so later on I mention how I attempt to minimise the variation caused by this choice.

Writing $f(\boldsymbol{X})$ as a linear model.

For each group, label the distinct vector values of $X_{\boldsymbol{G}_k}$ from 1 up to d_k. The node assigned the label d_k will correspond to the base value, but otherwise the labelling is irrelevant. For

group k, let the matrix $\boldsymbol{J}_k$ (size $n \times d_k$) indicate to which node each sample corresponds:

$$(\boldsymbol{J}_k)_{id} = \begin{cases} 1, & \text{if } X_{i\boldsymbol{G}_k} \text{ matches the } d\text{th node,} \\ 0, & \text{o/w.} \end{cases}$$

Define the matrix $\boldsymbol{J} = [\mathbf{1}\ \boldsymbol{J}_1^-\ \boldsymbol{J}_2^- \cdots \boldsymbol{J}_K^-]$, where $\mathbf{1}$ is a vector of ones and $\boldsymbol{J}_k^-$ represents the matrix $\boldsymbol{J}_k$ with the last column removed (which corresponds to setting θ_{kd_k} to zero). The underlying relationship can be written as the linear model $f(\boldsymbol{X}) = \boldsymbol{J}\boldsymbol{\Theta}$, where $\boldsymbol{\Theta} = \{\theta_0, \boldsymbol{\theta}_1^-, \boldsymbol{\theta}_2^-, \ldots, \boldsymbol{\theta}_K^-\}$, where $\boldsymbol{\theta}_k^-$ is the vector $\boldsymbol{\theta}_k$ with the last element removed. The degrees of freedom of this model is $D = 1 + \sum(d_k - 1)$, equal to the number of columns of $\boldsymbol{J}$ and the length of $\boldsymbol{\theta}$. Notice that the vector of regression coefficients completely defines the set of functions $\boldsymbol{f} = \{f_1, \ldots, f_K\}$. Therefore, a prior for the functions $\mathbb{P}(\boldsymbol{f}|\mathbb{G})$ can be specified in terms of a prior for the coefficients $\mathbb{P}(\boldsymbol{\Theta}|\mathbb{G})$.

Coefficient Prior $\mathbb{P}(\boldsymbol{\Theta}|\mathbb{G})$

$\mathbb{P}(\boldsymbol{\Theta}|\mathbb{G})$ is a joint distribution defined across the global intercept θ_0 and the coefficients θ_{kd} for $k = 1, 2, \ldots, K$ and $d = 1, 2, \ldots, d_k$. Because the labelling of nodes within groups is arbitrary, the (marginal) prior distributions for θ_{kd} must be the same for all d, while the arbitrary labelling of groups suggests identical priors for each $\boldsymbol{\theta}_k$ are appropriate.

Requiring the priors for elements within $\boldsymbol{\theta}_k$ to be identical might be viewed as an unfortunate consequence of the set-up. For example, in an association study we might wish to consider separately the contribution of marginal and interactive effects. To begin with, this would probably require a reformulation of each function. Returning to the case of a function of two binary predictors, suppose we were to define $f_k(0,0) = 0$, $f_k(1,0) = \theta_{k1}$, $f_k(0,1) = \theta_{k2}$ and $f_k(1,1) = \theta_{k1} + \theta_{k2} + \theta_{k3}$. Here, θ_{k3} would indicate the deviation from an additive model. We might choose to assign a smaller variance to θ_{k3}, if we felt its magnitude was likely to be less. Similarly, for the case of a single tertiary predictor, we might prefer values of $f_k(1)$ to be close to the midpoint of $f_k(0)$ and $f_k(2)$.

This is certainly an idea to think more about. Ideally, the user would have the choice to assign different variances for different types of effects. For the case of two binary or a single tertiary predictor, a possible reformulation is straightforward, but for more complex interactions, it does not appear so easy. Additionally, the current implementation benefits speed-wise from each sample corresponding to at most one node in each group, as this means that each row of $\boldsymbol{J}_k^-$ has at most one non-zero element. This advantage would likely have to be sacrificed. In any case, the version designed for quantitative predictors (Chapter 6) has an explicit preference for additivity built in, so is available to the user if they prefer.

The prior for θ_{kd} should reflect a preference for smaller effect sizes. This should offer protection against the presence of outliers, a danger to which a method such as *Sparse Partitioning* would otherwise be particularly vulnerable. Suppose the dth node of X_{G_k} is observed only once and corresponds to an extreme response value y_i. Setting $\theta_{kd} = y_i$ will likely result in a significant improvement in the model fit, providing "evidence" that the predictors in group k are associated. However, it is often more likely that an extreme value is a result of measurement error than a strong effect size. A prior which favours smaller regression coefficients will guard against this occurrence.

I decided to assign independent, identical, normal priors with mean zero to each coefficient. As well as having desirable characteristics, the normal distribution proves convenient later on when calculating marginal likelihoods. When the response is continuous, each prior has variance σ^2/r, where σ^2 corresponds to the variance of the residuals and is formally introduced when I explain the likelihood. When the response is binary, I set the variances to $1/r$. In both cases, the choice of r controls the extent coefficients are penalised on account of their size; larger values of r decrease the prior variance and therefore place greater emphasis on smaller-valued coefficients. Conversely, the user can decrease r, resulting in a less informative prior distribution.

The default value of r is 10, which for an association study, loosely speaking, supposes the contribution of each genetic effect is one order of magnitude lower than the residual noise. When analysing gene expression data, where the magnitudes of effects are likely to be higher, I suggest a smaller value for r, such as 1 or 2. To choose a more precise value of r, the user could consider the confidence interval implied by different settings. For example, when a continuous valued response has been standardised to have variance 1, setting r to 10 implies a 95% prior belief that effect sizes lie between approximately -0.6 and +0.6. Similar logic could be applied when the response is binary, considering a plausible range for odds ratios and selecting r accordingly. As with all Bayesian methods, if there is doubt when setting a prior parameter, it is prudent to repeat the analysis for a range of values and see the extent that the choice affects the results.

It is conceivable that the user might believe, for example, that there is one group of associations corresponding to strong effect sizes, while the other associations are more moderate. This could be reflected by introducing individual scaling terms r_k specific to each group. At present, my implementation of the MCMC sampling benefits speed-wise from the priors for each $\boldsymbol{\theta}_k$ being identical. Therefore, the value of such a change would depend on how strongly it was considered necessary.

2.5 Likelihood

When calculating the likelihood, *Sparse Partitioning* makes assumptions standard to generalized linear models, both with regards to the choice of link function and the distribution of the response values given the regression equation. In total, I consider three separate cases, as outlined in the following table:

Case	Response	Link Function
1	Continuous	Identity: $l(a) = a$.
2	Binary	Logit: $l(a) = \log(a/(1-a))$.
3	Binary	Probit: $l(a) = \Phi^{-1}(a)$.

(Φ is the cumulative density function for a standard normal distribution.)

Ideally, we wish to calculate the marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$ as this will, in conjunction with $\mathbb{P}(\mathbb{G})$, allow us to sample partitions from their posterior distribution. When the response is binary, this calculation is non-trivial, which is why two alternatives are offered.

2.5.1 Case 1: Continuous Response, Identity Link Function

When the response is continuous, the link function is the identity, so the regression equation becomes $\mathbb{E}(\boldsymbol{Y}) = f(\boldsymbol{X})$. The departures from the expected values, $Y_i - J_i\boldsymbol{\theta}$, are assumed to be independent draws from a normal distribution with mean zero and variance σ^2, leading to the following (raw) likelihood:

$$\mathbb{P}(\boldsymbol{Y}|f(\boldsymbol{X}), \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y} - f(\boldsymbol{X}))^T(\boldsymbol{Y} - f(\boldsymbol{X}))\right\},$$

or equivalently

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}, \sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{J\Theta})^T(\boldsymbol{Y} - \boldsymbol{J\Theta})\right\}.$$

σ^2 corresponds to the model noise, and therefore the variance which is not explained by the true underlying relationship. In genetic terms, this is proportional to one minus the (broad-sense) heritability, the percentage of observed variation contributable to genetic effects. In fact, as *Sparse Partitioning* standardises the response variance before analysis, the two are equal.

σ^2 is an unknown, so *Sparse Partitioning* assigns it a prior. As σ^2 represents the fraction of variance not explained, its prior distribution should be left-bounded at zero and heavily concentrated on the interval [0,1]. Copying the choice of *SSS*, which itself follows the reasoning of Dobra *et al.* (2004), I opted for $\mathbb{P}(\sigma^2) = \sigma^{-2}$. This density, a decreasing function, reflects a

preference for smaller values. It does not matter that this prior is improper, as I will explain later. One might favour a prior which peaks within the unit interval, if they have strong feelings concerning the proportion of variance explained. In this case, an inverse gamma prior could be used with nominal effect on any calculations.

Marginal Likelihood, $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G})$

As *Sparse Partitioning* is only concerned with the posterior distribution of partitions, σ^2, like $\boldsymbol{\Theta}$, is considered a nuisance parameter and it is convenient to remove both at the earliest instance. When the response is continuous, the marginal likelihood can be calculated explicitly:

$$\begin{aligned}\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) &= \int_{\boldsymbol{\Theta},\sigma^2} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G},\boldsymbol{\Theta},\sigma^2)\times\mathbb{P}(\boldsymbol{\Theta}|\mathbb{G})\times\mathbb{P}(\sigma^2)\,\mathrm{d}\boldsymbol{\Theta}\,\mathrm{d}\sigma^2\\ &= \int_{\boldsymbol{\Theta},\sigma^2} (2\pi\sigma^2)^{-\frac{n}{2}}\exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y}-\boldsymbol{J}\boldsymbol{\Theta})^T(\boldsymbol{Y}-\boldsymbol{J}\boldsymbol{\Theta})\right\}\\ &\quad\times(2\pi\sigma^2/r)^{-\frac{D}{2}}\exp\left\{-\tfrac{r}{2\sigma^2}\boldsymbol{\Theta}^T\boldsymbol{\Theta}\right\}\times\sigma^{-2}\,\mathrm{d}\boldsymbol{\Theta}\,\mathrm{d}\sigma^2.\end{aligned}$$

Letting $\boldsymbol{B}=\boldsymbol{J}^T\boldsymbol{J}+r\boldsymbol{I}_D$, where $\boldsymbol{I}_D$ is an identity matrix of size D, and $\boldsymbol{A}=\boldsymbol{B}^{-1}\boldsymbol{J}^T\boldsymbol{Y}$, we obtain

$$\begin{aligned}\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) &= r^{\frac{D}{2}}(2\pi)^{-\frac{n}{2}}\times\int_{\boldsymbol{\Theta}}(2\pi\sigma^2)^{-\frac{D}{2}}\exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{\Theta}-\boldsymbol{A})\boldsymbol{B}(\boldsymbol{\Theta}-\boldsymbol{A})\right\}\,\mathrm{d}\boldsymbol{\Theta}\\ &\quad\times\int_{\sigma^2}(\sigma^2)^{-(\frac{n}{2}+1)}\exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y}^T\boldsymbol{Y}-\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})\right\}\,\mathrm{d}\sigma^2\\ &= r^{\frac{D}{2}}(2\pi)^{-\frac{n}{2}}\times|\boldsymbol{B}|^{-\frac{1}{2}}\times\Gamma(\tfrac{n}{2})\,(\boldsymbol{Y}^T\boldsymbol{Y}-\boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})^{-\frac{n}{2}},\end{aligned}$$

where $\Gamma(\cdot)$ denotes the gamma function, and arises as the normalising constant of the gamma distribution. $\boldsymbol{J}^T\boldsymbol{J}$ will be a non-negative, symmetric matrix which can typically be inverted. The addition of $r\boldsymbol{I}_D$ ensures this is the case, although care must be taken for very small values of r as numerical inversion techniques may become unstable. Notice that the current prior choice for σ^2 has only a small impact on the marginal likelihood, only contributing 1 to the final power term. For standard sample sizes, this contribution is overwhelmed by $\frac{n}{2}$, suggesting the results of the method should be fairly robust to moderate changes to this prior.

In general, caution should be taken when using improper priors in Bayesian calculations. Suppose we wish to calculate a Bayes factor for Models 1 and 2, distinguished by the prior distributions $\mathbb{P}_1$ and $\mathbb{P}_2$, the second of which is improper. Although it might be possible to calculate $\mathbb{P}(\text{Data}|\text{Model 1})$, the integral of $\mathbb{P}(\text{Data}|\text{Parameters},\text{Model 1})\times\mathbb{P}_1$, and the same for Model 2, once we allow $\mathbb{P}_2$ to be unnormalised, we might just as reasonably have used $2\times\mathbb{P}_2$ instead. But doing this would double the evidence for the data under Model 2, unjustifiably adding support to the second model. However, for the case of $\mathbb{P}(\sigma^2)$, an improper prior is

permissible; σ^2 is used in all models, so its unnormalised value affects all posterior scores equally.

2.5.2 Case 2: Binary Response, Logit Link Function

When the response is binary, $\mathbb{E}(Y_i)$ equals $\mathbb{P}(Y_i = 1)$ and the regression equation becomes $l(\mathbb{P}(Y_i = 1)) = f(X_{i.})$. Therefore, the link function is required to convert a probability to a real value: $l(a)\colon [0,1] \to \mathbb{R}$. *Sparse Partitioning* is implemented for two choices of link function: a logit or a probit function.

Firstly, I consider the case of a logit link function, which takes the form $l(a) = \log(\frac{a}{1-a})$. The raw likelihood follows immediately by assuming that each response has been sampled independently, according to its probability of equalling 1:

$$\mathbb{P}(\boldsymbol{Y}|f(\boldsymbol{X})) = \prod_i [l^{-1}(f(X_i))]^{Y_i}\ [1 - l^{-1}(f(X_i))]^{(1-Y_i)},$$

or equivalently

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) = \prod_i [l^{-1}(J_i\boldsymbol{\Theta})]^{Y_i}\ [1 - l^{-1}(J_i\boldsymbol{\Theta})]^{(1-Y_i)}.$$

Marginal Likelihood, $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$

When a logit link function is used, *Sparse Partitioning* estimates the marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$ through use of a Laplace approximation (de Bruijn, 1958). First the method calculates the posterior mode of $\boldsymbol{\Theta}$ given $\mathbb{G}$, then it applies an approximation centred on this value. Let $W(\boldsymbol{\Theta}) = \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \times \mathbb{P}(\boldsymbol{\Theta}|\mathbb{G})$, so that the marginal likelihood equals the integral of $W(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Theta}$. Additionally, let $w(\boldsymbol{\Theta}) = \log(W(\boldsymbol{\Theta}))$. By applying Taylor's theorem about a value $\boldsymbol{\Theta}'$, we obtain

$$w(\boldsymbol{\Theta}) \approx w(\boldsymbol{\Theta}') + (\boldsymbol{\Theta} - \boldsymbol{\Theta}')^T \frac{\mathrm{d}w(\boldsymbol{\Theta}')}{\mathrm{d}\boldsymbol{\Theta}} + \frac{1}{2}(\boldsymbol{\Theta} - \boldsymbol{\Theta}')^T \frac{\mathrm{d}^2 w(\boldsymbol{\Theta}')}{\mathrm{d}\boldsymbol{\Theta}^2} (\boldsymbol{\Theta} - \boldsymbol{\Theta}').$$

If $\boldsymbol{\Theta}' = \hat{\boldsymbol{\Theta}}$, where $\hat{\boldsymbol{\Theta}}$ is the argument of the maximum of $w(\boldsymbol{\Theta})$, then

$$W(\boldsymbol{\Theta}) \approx W(\hat{\boldsymbol{\Theta}}) \exp\left\{ -\frac{1}{2}(\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}})^T \left(-\frac{\mathrm{d}^2 w(\hat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}^2} \right) (\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}) \right\}.$$

Therefore

$$\begin{aligned}\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) &= \int_{\boldsymbol{\Theta}} W(\boldsymbol{\Theta})\,\mathrm{d}\boldsymbol{\Theta} \\ &\approx W(\hat{\boldsymbol{\Theta}})(2\pi)^{\frac{D}{2}}\left|-\frac{\mathrm{d}^2 w(\hat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}^2}\right|^{-\frac{1}{2}} \\ &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G},\hat{\boldsymbol{\Theta}})\times\mathbb{P}(\hat{\boldsymbol{\Theta}}|\mathbb{G})(2\pi)^{\frac{D}{2}}\left|-\frac{\mathrm{d}^2 w(\hat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}^2}\right|^{-\frac{1}{2}}.\end{aligned}$$

This approximation requires the calculation of $\hat{\boldsymbol{\Theta}}$, the posterior mode of $\boldsymbol{\Theta}$ for a particular partition. Being the root of $w'(\boldsymbol{\Theta}) = \frac{\mathrm{d}}{\mathrm{d}\Theta}w(\boldsymbol{\Theta})$, it can be estimated using the Newton-Raphson method, whose history is traced in Ypma (1995). This method prescribes an iterative procedure, based on repeated approximations of $w'(\boldsymbol{\Theta})$ about $\hat{\boldsymbol{\Theta}}$.

$$0 = w'(\hat{\boldsymbol{\Theta}}) \approx w'(\boldsymbol{\Theta}) + (\hat{\boldsymbol{\Theta}}-\boldsymbol{\Theta})^T w''(\boldsymbol{\Theta}) \;\Rightarrow\; \boldsymbol{\Theta}^{t+1} = \boldsymbol{\Theta}^t - (w''(\boldsymbol{\Theta}^t))^{-1}\,w'(\boldsymbol{\Theta}),$$

where $\boldsymbol{\Theta}^1, \boldsymbol{\Theta}^2, \ldots, \boldsymbol{\Theta}^t, \boldsymbol{\Theta}^{t+1}, \ldots$ are a series of realisations of $\boldsymbol{\Theta}$. In our case, the required derivatives can be calculated explicitly:

$$w(\boldsymbol{\Theta}) = \sum_i Y_i \log p_i + (1-Y_i)\log(1-p_i) - \frac{r}{2}\boldsymbol{\Theta}^T\boldsymbol{\Theta} - \frac{D}{2}\log(2\pi/r),$$

where $p_i = (1-\exp(-J_i\boldsymbol{\Theta}))^{-1}$. Making use of

$$\frac{\mathrm{d}}{\mathrm{d}\Theta_j}p_i = \frac{\exp(-J_i\boldsymbol{\Theta})}{(1+\exp(-J_i\boldsymbol{\Theta}))^2}J_{ij} = p_i(1-p_i)J_{ij},$$

we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\Theta_j}w(\boldsymbol{\Theta}) &= \sum_i Y_i(1-p_i)J_{ij} - (1-Y_i)p_iJ_{ij} - r\Theta_j \\ &= \sum_i (Y_i-p_i)J_{ij} - r\Theta_j\end{aligned}$$

and

$$\frac{\mathrm{d}^2}{\mathrm{d}\Theta_j\mathrm{d}\Theta_k}w(\boldsymbol{\Theta}) = \sum_i -p_i(1-p_i)J_{ij}J_{ik} - r\mathbf{1}(j=k),$$

where $\mathbf{1}(\cdot)$ represents an indicator function taking value 1 or 0 according to whether its argument is true or false. These derivatives allow calculation of the Newton-Raphson iterations. The user specifies a tolerance value, which determines whether $\boldsymbol{\Theta}^{t+1}$ is "considered equal" to $\boldsymbol{\Theta}^t$, in which case the iterations stop. There is no guarantee that the Newton-Raphson iterations converge, but so far I have always found this to be the case. However, should convergence

become an issue, an alternative algorithm could be implemented, line-search or otherwise (this approach is discussed in the supplement of Marchini *et al.*, 2007).

2.5.3 Case 3: Binary Response, Probit Link Function

Instead of a logit function, *Sparse Partitioning* is set up to permit use of a probit link. This takes the form $l(a) = \Phi^{-1}(a)$, where Φ is the cumulative density function of a standard normal distribution.

The raw likelihood takes the same form as for Case 2, determined by supposing each response is sampled at random, according to its probability of equalling 1:

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) = \prod_i [l^{-1}(J_i\boldsymbol{\Theta})]^{Y_i} \ [1 - l^{-1}(J_i\boldsymbol{\Theta})]^{(1-Y_i)}.$$

When a probit link function is used, the Laplace approximation is no longer tractable so can not be employed to calculate the marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$. Therefore, *Sparse Partitioning* takes a different approach, introducing latent variables based on the set-up proposed by Albert and Chib (1993). This creates a new set of variables $\boldsymbol{Z} = \{Z_1, Z_2, \ldots, Z_n\}$, where each component acts as an intermediary between a sample's underlying relationship value and its response. Each Z_i is distributed according to a truncated normal distribution with mean $J_i\boldsymbol{\Theta}$, variance 1 and domain determined by the value of the corresponding response:

$$\mathbb{P}(Z_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \propto \mathbb{N}(J_i\boldsymbol{\Theta}, 1) \text{ with } \begin{cases} Z_i > 0, & \text{if } Y_i = 1, \\ Z_i \leq 0, & \text{if } Y_i = 0, \end{cases}$$

where $\mathbb{N}(a, b)$ represents a normal distribution with mean a and variance b. It is easy to see that this leads to the required link function, as it results in $\mathbb{P}(Y_i = 1) = \Phi(J_i\boldsymbol{\Theta})$:

$$\begin{aligned}
\mathbb{P}(Y_i = 1|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) &= \int_{Z_i} \mathbb{P}(Y_i = 1|Z_i, \boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \times \mathbb{P}(Z_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \ \mathrm{d}Z_i \\
&= \int_{Z_i} \mathbb{P}(Y_i = 1|Z_i) \times \mathbb{P}(Z_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \ \mathrm{d}Z_i \\
&= \int_{Z_i} \mathbf{1}(Z_i > 0) \times \mathbb{P}(Z_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \ \mathrm{d}Z_i \\
&= \int_{Z_i > 0} \mathbb{P}(Z_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}) \ \mathrm{d}Z_i \\
&= \mathbb{P}(Z_i > 0|Z_i \sim \mathbb{N}(J_i\boldsymbol{\Theta}, 1)) \\
&= \mathbb{P}(Z_i' > -J_i\boldsymbol{\Theta}|Z_i' \sim \mathbb{N}(0, 1)) \\
&= \Phi(J_i\boldsymbol{\Theta}).
\end{aligned}$$

Marginal Likelihood, $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})$

Albert and Chib's decision to introduce latent variables was motivated by a desire to sample from $\boldsymbol{\Theta}$. Typically, their formulation results in a known form for, and thus easy sampling from, the conditional posterior distribution of elements of $\boldsymbol{\Theta}$. Although *Sparse Partitioning* is not interested in sampling from $\boldsymbol{\Theta}$, a consequence of introducing $\boldsymbol{Z}$ is that it becomes possible to calculate the marginal likelihood $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})$ explicitly. This calculation follows the same steps used to find $\mathbb{P}(Y|\boldsymbol{X},\mathbb{G},\boldsymbol{\Theta})$ in Case 1, except that now σ^2 is fixed at 1:

$$\begin{aligned}
\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G}) &= \int_{\boldsymbol{\Theta}} \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G},\boldsymbol{\Theta}) \times \mathbb{P}(\boldsymbol{\Theta}|\mathbb{G})\ \mathrm{d}\boldsymbol{\Theta} \\
&= \int_{\boldsymbol{\Theta}} (2\pi)^{-\frac{n}{2}} \exp\left\{-\tfrac{1}{2}(\boldsymbol{Z}-\boldsymbol{J\Theta})^T(\boldsymbol{Z}-\boldsymbol{J\Theta})\right\} \times (2\pi/r)^{-\frac{D}{2}} \exp\left\{-\tfrac{1}{2}\boldsymbol{\Theta}^T\boldsymbol{\Theta}\right\} \\
&= r^{\frac{D}{2}}(2\pi)^{-\frac{n}{2}} \times |\boldsymbol{B}|^{-\frac{1}{2}} \times \exp\left\{-\tfrac{1}{2}(\boldsymbol{Z}^T\boldsymbol{Z}-\boldsymbol{A}^T\boldsymbol{BA})\right\}.
\end{aligned}$$

Calculation of $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})$ proves useful later on. When using a probit link function, instead of seeking $\mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{Y})$ directly, *Sparse Partitioning* samples from $\mathbb{P}(\mathbb{G},\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{Y})$, from which the former can be obtained. This relies upon a combination of Gibbs' and Metropolis-Hastings theory, alternately drawing from $\mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$ and $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{Y},\mathbb{G})$, both of which require calculation of $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})$.

An alternative approach is possible, one which continues with the latent variable representation, but instead tries to estimate the marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G})$ directly.

$$\begin{aligned}
\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) &= \int_{\boldsymbol{Z}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})\ \mathrm{d}\boldsymbol{Z} \\
&= \int_{\boldsymbol{Z}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z}) \times \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})\ \mathrm{d}\boldsymbol{Z} \\
&= \int_{\boldsymbol{Z}\in\boldsymbol{Z}^\dagger} r^{\frac{D}{2}}(2\pi)^{-\frac{n}{2}} \times |\boldsymbol{B}|^{-\frac{1}{2}} \times \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Z}^T\boldsymbol{Z}-\boldsymbol{A}^T\boldsymbol{BA})\right\}\ \mathrm{d}\boldsymbol{Z} \\
&\propto \int_{\boldsymbol{Z}\in\boldsymbol{Z}^\dagger} \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Z}^T(\boldsymbol{I}_n-\boldsymbol{JB}^{-1}\boldsymbol{J}^T)\boldsymbol{Z})\right\}\ \mathrm{d}\boldsymbol{Z},
\end{aligned}$$

where $\boldsymbol{Z}^\dagger$ is the n-dimensional domain over which $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z})=1$:

$$\boldsymbol{Z}\in\boldsymbol{Z}^\dagger \Leftrightarrow \begin{cases} Z_i > 0, & \text{if } Y_i = 1, \\ Z_i \leq 0, & \text{if } Y_i = 0. \end{cases}$$

Within the integrand, $\boldsymbol{Z}$ takes the form of a multivariate normal distribution $\mathbb{N}(0,\boldsymbol{C}^{-1})$, where $\boldsymbol{C}=\boldsymbol{I}_n-\boldsymbol{JB}^{-1}\boldsymbol{J}^T$. Let $\boldsymbol{E}$ denote the Cholesky decomposition of $\boldsymbol{C}$, such that $\boldsymbol{C}=\boldsymbol{E}^T\boldsymbol{E}$, then the distribution of $\boldsymbol{EZ}$ will also be normal: $\boldsymbol{EZ}\sim\mathbb{N}(0,\boldsymbol{I}_n)$. Crucially, each $\boldsymbol{EZ}$ is independent, so the marginal likelihood can be written as a product of integrals across the

new region $\boldsymbol{EZ}^{\dagger}$. Analytic methods exist for calculating this integral (Curnow and Dunnett, 1962). However, my feeling was that the benefits of being able to calculate $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$ directly would be overshadowed by the computational demands and approximate nature of this approach.

For all three link functions, the user can choose to replace the marginal likelihood with the maximum likelihood estimate (this choice is automatic when r is set to zero, as then the priors on effect sizes become improper). This change gives the method a more frequentist feel and slightly faster run time. However, I almost always consider this approach inferior to the fully Bayesian version and advise against its use.

Discussion: Independence of Partition and Function Priors

Earlier on, I mentioned that the function prior depends loosely on the partitions. The reason for this is that each partition defines the structure of the linear model representation $f(\boldsymbol{X}) = \boldsymbol{J\Theta}$ and therefore the number of regression coefficients. In a similar manner, the function also depends on $\boldsymbol{X}$ as, given a partition, its degrees of freedom is determined by the number of nodes observed for each group.

The latter dependency is easy to remove. For example, consider a partition containing a non-null group of size two. *Sparse Partitioning* caters for tertiary predictors, so will allow for up to 9 possible nodes; but if the predictors are binary, at most four of these will be present. Suppose additional coefficients, corresponding to the unobserved nodes, are added to the regression model and assigned identical priors. This will remove the dependency of the function on the observed value of $\boldsymbol{X}$. However, these additional coefficients will not interfere with the raw likelihood and can be integrated out immediately, so leaving the marginal likelihood unaffected.

A similar trick could be used to remove the dependency of the function prior on the current partition. In theory, we could specify a linear model containing a set of regression coefficients relating to every possible partition. All coefficients, except those related to the current partition, could be integrated out immediately with no effect on the marginal likelihood.

This logic comes in useful later on, when considering how to sample from the posterior distribution. Because the length of $\boldsymbol{\Theta}$ is determined by the partition, one might assert that "Reversible Jump" MCMC is required to account for the changes in dimension. However, this is not the case if we consider that the dimension stays constant and only the number of active coefficients varies. *Sparse Partitioning* takes advantage of the way Bayesian methods enforce Occam's Razor (Murray and Ghahramani, 2005; Ghahramani, 2010), a principle which

insists that for two equally well fitting models, the more simple one should be preferred. In the case of *Sparse Partitioning*, simplicity depends, in part, on the number of active regression coefficients. Therefore, the additional coefficients, which do not contribute to the underlying relationship, should not contribute to the complexity penalty, as evidenced by them integrating to unity.

Discussion: Choices of Link Function

Recall that the regression equation takes the form $l(\mathbb{E}(\boldsymbol{Y})) = f(\boldsymbol{X})$. The raw likelihood is calculated by comparing how well each observed response value Y_i agrees with its predicted value $l^{-1}(f(X_i))$. The link function determines how errors in specifying $f(X_i)$ are carried through to the likelihood. In the continuous response case, symmetry seems desirable, so that underestimation of $l^{-1}(f(X_i))$ is penalised equally to overestimation, and the same for $f(X_i)$. The former is automatically true on account of the choice of likelihood function; the normal assumption considers only the magnitude of the residuals. To ensure the latter, the symmetry must be maintained by the link function. To obtain such a function, its curve-representation should possess symmetry about all points. This implies it is linear and therefore, without loss of generality, the identity.

For a binary response, the choice of link function seems less intuitive. As $l^{-1}(f(X_i))$ corresponds to a probability, its value must map to the interval [0,1]. Penalising misspecification symmetrically seems no longer sensible, as when, say, the true value of $\mathbb{E}(Y_i)$ is 0.8, the predicted value can be four times too small as it can be too large. If forced to choose between the logit and probit functions, the latter seems more readily justified. It supposes that a sample's response is governed by an underlying normal distribution, but in effect all we are able to observe is whether the value is positive ($Y = 1$) or negative ($Y = 0$). By contrast, results using the logit link function are more easily interpreted. The regression coefficients $\boldsymbol{\Theta}$ correspond to the log odds, or equivalently $\exp(\boldsymbol{\Theta})$ indicates how much a unit change in each of the predictors affects the odds ratio.

Reassuringly, I feel that in nonlinear regression, the risk of choosing a poor link function is reduced. When predictors are forced to contribute linearly to the underlying relationship, the link function determines how much a unit change of each predictor affects the expected value of the response, so an incorrect link function will have a direct knock-on effect. This is not the case for nonlinear underlying relationships. Their flexibility to allow predictors to contribute in a nonlinear manner means that when the link function is inaccurate, to some extent this error can be compensated for.

As Figure 2.5.3 demonstrates, the shapes of the (inverse) logit and probit functions are

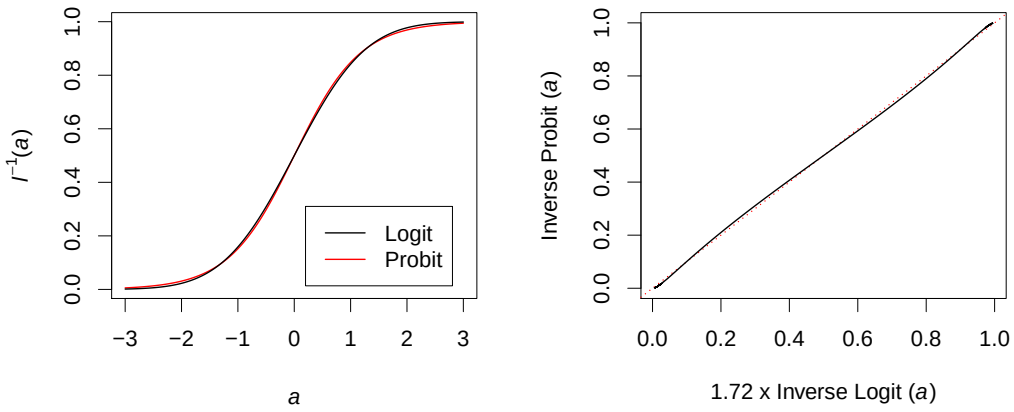


Figure 2.3: *Comparison of the logit and probit link functions. The two plots provide a comparison of the (inverse) logit and probit functions, two link functions suitable for converting a real value to a probability. The left plot overlays the functions. That the two lines are almost indistinguishable, demonstrates how closely the functions agree, especially for mid-range values of a. The right graph matches the images of each, plotting $l^{-1}(a)$ for the logit and probit functions across the range of a. This time, the line barely deviates from the diagonal, once more showing the similarity of the two link functions.*

very similar. For these plots, I rescaled the logit function to better highlight the concordance; the two curves plotted are in fact $l^{-1}(a) = \Phi(a)$ and $l^{-1}(a) = (1 + \exp(-1.72a))^{-1}$. This scaling should not detract from the comparison, as it is equivalent to scaling $\boldsymbol{\Theta}$, which in *Sparse Partitioning* can be achieved by altering $\boldsymbol{\Theta}$'s prior variance.

Therefore, supposing that both the logit and probit functions provide a reasonable choice, it remains to decide which is preferable. Generally, it is better to integrate across nuisance parameters, as this averages over their uncertainty and reduces the complexity of the model space. As the calculation of the posterior estimates will be based on MCMC sampling, which attempts to obtain a fair sampling across all parameters, the fewer parameters we have, the better. This logic would suggest a logit function should be chosen, as then *Sparse Partitioning* calculates $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$ directly. However, the issue here is that the Laplace approximation still introduces inaccuracies. Furthermore, its calculation is time consuming due to the Newton-Raphson iterations. Therefore, I continue to describe the method for both a logit and probit function, then return to this debate when testing each implementation.

Discussion: Identifiability of Functions

To ensure identifiability, a global intercept θ_0 was introduced, and one node from each group was picked as the base value and mapped to zero. The calculations of $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$ or $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X}, \mathbb{G})$ will vary depending on which nodes are chosen to act as the base values for each function. In order to reduce these inconsistencies, *Sparse Partitioning* chooses these base nodes according to a defined rule, which when possible picks the zero vector of $X_{\boldsymbol{G}_k}$. Additionally, continuous

response values are first transformed to have mean 0 and variance 1, which reduces the variability resulting from different choices of nodes.

An alternative way to ensure identifiability is to retain the global intercept, but insist that each group of regression coefficients sums to zero: $\sum_{d=1}^{d_k} \theta_{kd} = 0$, for $k = 1, 2, \ldots, K$. This condition might appear preferable, as it affects the fit of all coefficients rather than just one. Superficially, it would require very little extra work; before discarding the final column of $\boldsymbol{J}_k$, its value would first be subtracted from each remaining column. However, *Sparse Partitioning* is able to utilise the fact that $\boldsymbol{J}_k^-$ has no more than one non-zero element in each row. While the resulting speed-up is minimal with tertiary predictors, when applied to non-tertiary values, it proves considerable.

For the most part, *Sparse Partitioning* does not actually require identifiability; the method is not interested in obtaining the posterior distribution of $\boldsymbol{\Theta}$ and all calculations described will be possible even if the final column of $\boldsymbol{J}_k$ is retained. The only exception to this rule is when r is set to zero, in which case the prior on the regression coefficients becomes improper and *Sparse Partitioning* is forced to replace the marginal likelihood with the maximum likelihood value. As it stands, this option is redundant. If chosen, the functions will no longer be penalised according to complexity, so the method will opt for the most complicated ones possible. However, it becomes (slightly) relevant when considering the extension to quantitative predictors (Chapter 6).

Comparison with *BEAM*

The partitioning concept of *Sparse Partitioning* has aspects in common with *BEAM* (Zhang and Liu, 2007). The latter looks to group the predictors into three classes: those not associated, those which contribute additively and those which contribute jointly. Zhang and Liu obtain a score for each partition by considering $\mathbb{P}(\boldsymbol{X}|\boldsymbol{Y}, \mathbb{G})$, the likelihood of the predictors given the response values and the groupings. Their premise is that in most instances, a predictor's underlying state frequencies will be the same across all samples; but for the associated predictors, there will be two sets of frequencies, one corresponding to cases ($Y_i = 1$) and one to controls ($Y_i = 0$). *BEAM*'s model can be considered nested within that of *Sparse Partitioning*, and could be implemented by forcing all of the groups except one to be singleton. With this set-up, a model's degrees of freedom will necessarily grow exponentially with the number of predictors involved in interactions. This increase will be faster than for *Sparse Partitioning* (which employs a combination of linear and exponential growth) and will likely restrict further the number of associations that could feasibly be considered.

In a sense, *BEAM*'s approach operates in the correct direction. Although we expect an

individual's phenotype to be influenced by its genetic variants, in most cases the samples will have been chosen according to outcome, so a retrospective likelihood is more appropriate. Prentice and Pyke (1979) demonstrate the equivalence of the two approaches (for a simple model) within a frequentist framework, while Seaman and Richardson (2004) draw similar conclusions in a Bayesian set-up. This is fortunate, as there are many advantages to a prospective model. One of these is the ease with which confounding and missing values can be included, which I discuss in Chapter 3.

In Bayesian methods, it is typically a great advantage to be able to calculate the marginal likelihood explicitly, as resorting to numerical solutions will have a detrimental effect on accuracy and lead to an increased run time. A sufficient condition for explicit calculation of the marginal likelihood, is that the raw likelihood equation $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}, \ldots)$ can be broken down as the product of likelihoods for groups of response values, each of which involves only a single coefficient of $\boldsymbol{\Theta}$ (and no coefficient is involved for more than one group). If this condition holds, and assuming conjugate priors are used, the marginal likelihood can be obtained explicitly by integrating across each coefficient individually. In *Sparse Partitioning*, this condition is not true, as most response values will be affected by one coefficient from each group. More generally, for a prospective model, this condition can not hold once more than one group of associated predictors are permitted ($K > 1$), nor as soon as confounding variables are introduced (as these variables will affect all samples). To a large extent, this explains why so few methods allow multiple groups or consider confounding, and is perhaps why the method *BAMSE* (Albrechtsen *et al.*, 2007), discussed in the introduction, forces samples to belong to only one risk set. (Notice that this condition is not a necessary one, as in *Sparse Partitioning* conjugacy exists when the response is continuous, owing to properties of the normal distribution.)

In *BEAM*, this condition holds (although, because a retrospective model is used, the condition applies instead to $\mathbb{P}(\boldsymbol{X}|\boldsymbol{Y}, \mathbb{G}, \boldsymbol{\Theta})$). Fast computation of the marginal likelihood proves a huge advantage, and allows the method to consider problems involving over 100,000 predictors. Therefore, if the restriction to the underlying relationship is not a concern, nor are confounding variables an issue, when presented with case-control data, this method is appealing.

Intuitively, continuous response variables should be more informative than binary ones, and be more conducive for the search for interactions. However, there appears to be no obvious way to adapt *BEAM* for a continuous response. The retrospective approach causes difficulties. For the predictors declared associated, it would be necessary to devise a distribution of state frequencies given a continuous value. For tertiary predictors, such a distribution should have 2 degrees of freedom, which can not be obtained from a one dimensional response. When considering SNP values, one could assume the underlying state frequencies for associated pre-

dictors obey the ratio $p^2 : 2p(1-p) : (1-p)^2$ (Hardy, 1908; Weinberg, 1908) and hence use a logit or probit link function to connect p_i to Y_i. But, considering that for associated SNPs the distribution of their alleles will depend on the phenotype, such an assumption is perhaps unlikely to be realistic.

Since the original submission of this thesis, a version of *BEAM* suitable for continuous response values has been proposed (Zhang *et al.*, 2011). However, to do so, the authors have abandoned the retrospective approach. Secondly, they have removed the option for predictors to contribute additively, instead considering only the most general model where all associated predictors interact with each other. They allocate a mean value to each vector node, to which they assign a normal prior. Being a draft version, there is no mention yet of implementation, however, the proposed method appears very similar to running *Sparse Partitioning* with K set to 1, save for the option to include confounding variables.

2.6 Posterior Distribution

For Case 1 (continuous response, identity link function) and Case 2 (binary response, logit link function), we wish to find $\mathbb{P}(\mathbb{G}|\boldsymbol{X}, \boldsymbol{Y})$. For Case 3 (binary response, probit link function), we wish to find $\mathbb{P}(\mathbb{G}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y})$. For all cases, explicit calculation of the posterior would require an exhaustive search across partitions, which would not be possible. To give an idea of the magnitude of this task, consider that the number of unique sets of associations, and therefore the number of equivalence classes, equals 2^N, and thus grows exponentially with the number of predictors. Even were each equivalence class to have only one member, it would still not be feasible to visit all partitions unless N was very small.

Therefore, *Sparse Partitioning* always resorts to MCMC sampling, creating a Markov chain starting at the null partition $\mathbb{G}^0$ which declares no predictors associated. The following table outlines, for each case, the stages involved in one iteration of this sampling:

Case	Target Posterior	Sampling Stages
1	$\mathbb{P}(\mathbb{G}\|\boldsymbol{X}, \boldsymbol{Y})$	1 - Sample $\boldsymbol{I}$ | 2 - Sample $G_{k,j}$
2	$\mathbb{P}(\mathbb{G}\|\boldsymbol{X}, \boldsymbol{Y})$	1 - Sample $\boldsymbol{I}$ | 2 - Sample $G_{k,j}$
3	$\mathbb{P}(\mathbb{G}, \boldsymbol{Z}\|\boldsymbol{X}, \boldsymbol{Y})$	1 - Sample $\boldsymbol{I}$ | 2 - Sample $G_{k,j}$ | 3 - Sample $\boldsymbol{Z}$

MCMC sampling requires frequent calculation of a model's (relative) posterior weighting. Therefore, it proves convenient to compute a posterior "score" for each partition, proportional

to its conditional posterior. Notice that in Case 3, this value will not depend on $\boldsymbol{Y}$:

$$\begin{aligned}\mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z}) &= \mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{Z}) \times \mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{X},\mathbb{G})/\mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{X}) \\ &= \mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{Z}) \times \mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z})/\mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z}) \\ &= \mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{Z}).\end{aligned}$$

Therefore, we can define

$$\text{Score}(\mathbb{G}) := \begin{cases} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G}), & \text{for Cases 1 and 2,} \\ \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G}), & \text{for Case 3.} \end{cases}$$

This score is not only involved when considering a change to the current partition. For every single move, it plays a part in determining either the proposal distribution or the acceptance probability of the proposal. $\mathbb{P}(\mathbb{G})$ is calculated in advance and stored in a look-up table, so the time taken to score a partition, and therefore the overall algorithm run time, is dictated by how long it takes to compute the marginal likelihood.

2.6.1 Stage One: Sampling each Component of $\boldsymbol{I}$

I explain this stage supposing we are interested in Cases 1 or 2. For Case 3 the only difference is that $\boldsymbol{Y}$ is replaced by $\boldsymbol{Z}$ in all steps.

Sparse Partitioning begins each iteration by sampling new values for each element of $\boldsymbol{I} = (I_1, I_2, \ldots, I_N)$, the vector indicating the group membership of each predictor. For each I_g, the method uses a proposal distribution which matches the element's conditional posterior distribution: $\mathbb{Q}(I_g|I_{-g}) = \mathbb{P}(I_g|I_{-g}, \boldsymbol{X}, \boldsymbol{Y})$. By choosing this proposal distribution, the acceptance rate will always be 1, so the sampled value can be accepted immediately (Gibbs' Sampling). To guard against the introduction of order-based biases, the sequence in which each I_g is sampled is randomised for each iteration.

The proposal distribution for selecting a new value I_g^* can be calculated explicitly by scoring all partitions that differ from the current partition only in their value of I_g:

$$\begin{aligned}\mathbb{Q}(I_g^*) = \frac{\mathbb{P}(I_g^*|I_{-g},\boldsymbol{X},\boldsymbol{Y})}{\sum_{I_g'}\mathbb{P}(I_g'|I_{-g},\boldsymbol{X},\boldsymbol{Y})} &= \frac{\mathbb{P}(I_g^*,I_{-g}|\boldsymbol{X},\boldsymbol{Y})}{\sum_{I_g'}\mathbb{P}(I_g',I_{-g}|\boldsymbol{X},\boldsymbol{Y})} \\ &= \frac{\text{Score}(I_g^*,I_{-g})}{\sum_{I_g'}\text{Score}(I_g',I_{-g})}.\end{aligned}$$

The order that these partitions are searched mimics the way the truncated Bell numbers are calculated. First, predictor g is removed from the current partition. Then, it is added, in turn,

to the null group, each non-empty non-null group (if space) and then as a singleton non-null group (if space).

This system ensures that invalid partitions, those with zero prior weight, are not considered. For example, when all predictors are in the null group, any non-zero value for I_g will result in the same partitioning. The prior distribution reflects this fact by assigning prior weight only to the case $I_g = 1$.

When $C > 1$, each copy is treated as a new predictor, so the number of samplings required in this stage increases by a factor of C. To combat this increase, the order that each I_g is sampled is no longer completely random. Instead, elements of $\boldsymbol{I}$ which correspond to copies of the same predictor are sampled consecutively. I took this decision for computational reasons. If the current partition does not declare associated any copy of a particular predictor, then $\mathbb{Q}(I_g^*)$ will be the same for all copies. As long as each copy remains not associated, there is no need to recalculate this distribution. As a result, for sparse problems, increasing C will have minimal effect on the time this sampling stage takes.

2.6.2 Stage Two: Sampling a Component of $\mathbb{G}$

Once more, the description of this stage differs only slightly between Cases 1 or 2 and Case 3. For the latter, simply replace $\boldsymbol{Y}$ with $\boldsymbol{Z}$ in all instances.

Sparse Partitioning next picks at random a predictor from $S_{\boldsymbol{I}}$, the set of predictors currently associated. Suppose this predictor corresponds to element j of group k. The method resamples G_{kj} from its conditional posterior distribution: $\mathbb{Q}(G_{kj}) = \mathbb{P}(G_{kj}|\mathbb{G}^{-kj}, \boldsymbol{X}, \boldsymbol{Y})$, where $\mathbb{G}^{-kj}$ denotes the current partition $\mathbb{G}$ with component G_{kj} removed. Again, this distribution is calculated exhaustively, by first removing the component, then testing all possibilities in its place:

$$\mathbb{Q}(G_{kj}^*) = \frac{\mathbb{P}(G_{kj}^*|\mathbb{G}^{-kj}, \boldsymbol{X}, \boldsymbol{Y})}{\sum_{G'_{kj}} \mathbb{P}(G'_{kj}|\mathbb{G}^{-kj}, \boldsymbol{X}, \boldsymbol{Y})} = \frac{\mathbb{P}(G_{kj}^*, \mathbb{G}^{-kj}|\boldsymbol{X}, \boldsymbol{Y})}{\sum_{G'_{kj}} \mathbb{P}(G'_{kj}, \mathbb{G}^{-kj}|\boldsymbol{X}, \boldsymbol{Y})}$$
$$= \frac{\text{Score}(G_{kj}^*, \mathbb{G}^{-kj})}{\sum_{G'_{kj}} \text{Score}(G'_{kj}, \mathbb{G}^{-kj})}.$$

When $C > 1$, the conditional posterior probability is the same for each available copy of a predictor. Therefore, as with the first sampling stage, it is only necessary to calculate the score for one copy and increasing C has minimal effect on computation time.

2.6.3 Stage Three: Sampling each Component of $\boldsymbol{Z}$

As a reminder, this stage is only used for Case 3.

When sampling new elements of $\boldsymbol{Z}$, we are unable to use Gibbs' sampling. If $\boldsymbol{\Theta}$ remained in the model, we could sample directly from the conditional posterior of Z_i, which follows immediately from its definition. As this is not the case, we would have to consider the conditional posterior distribution once $\boldsymbol{\Theta}$ has been integrated out.

$$\begin{aligned}\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\boldsymbol{Y},\mathbb{G}) &\propto \mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z},\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G}) \\ &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z}) \times \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G}) \\ &\propto \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Z}^T(\boldsymbol{I}_n + \boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^T)\boldsymbol{Z})\right\}.\end{aligned}$$

To appreciate the last line, observe that the domain of $\boldsymbol{Z}$ is defined as all values such that $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{Z}) = 1$, while the expression for $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X},\mathbb{G})$ follows from previous calculations. Although this tells us that the joint conditional posterior distribution of $\boldsymbol{Z}$ is truncated normal, I can not see an efficient means to sample from its components; neither are they independent, nor can we integrate with respect to Z_{-i}.

Therefore, *Sparse Partitioning* resorts to proposing a new value of Z_i from a folded standard normal distribution, with its sign determined by Y_i:

$$\mathbb{Q}(Z_i^*) = 2\,\phi(Z_i^*), \text{ where } \begin{cases} Z_i > 0, & \text{if } Y_i = 1, \\ Z_i \leq 0, & \text{if } Y_i = 0, \end{cases}$$

where ϕ represents the density function of a standard normal distribution. A proposed value Z_i^* is accepted with probability $\min(1, \alpha)$, where

$$\alpha = \frac{\mathbb{P}(\mathbb{G}, Z_i^*, Z_{-i}|\boldsymbol{X},\boldsymbol{Y})}{\mathbb{P}(\mathbb{G}, Z_i, Z_{-i}|\boldsymbol{X},\boldsymbol{Y})} \times \frac{\phi(Z_i)}{\phi(Z_i^*)}.$$

With gentle rearrangement, we obtain

$$\alpha = \frac{\mathbb{P}(Z_i^*, Z_{-i}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G})}{\mathbb{P}(Z_i, Z_{-i}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G})} \times \frac{\phi(Z_i)}{\phi(Z_i^*)},$$

the denominator of the first fraction being the score of the current partition, while the numerator is the score when Z_i is replaced by Z_i^*.

2.6.4 Obtaining Posterior Estimates

The user specifies the number of iterations performed by *Sparse Partitioning*. For diagnostic reasons, which I discuss later, the first quarter of the Markov Chain or 1000 iterations, whichever is shorter, is treated as a burn-in period and discarded. The remaining iterations are divided equally into two sections. Final posterior estimates are calculated from both sections combined, however, estimates from each section individually can be compared in order to assess convergence.

To obtain each set of estimates, a tally is kept of how often each predictor is declared associated. Additionally, the method keeps track of pairs of predictors which appear together. To avoid unnecessary memory allocation, *Sparse Partitioning* only creates a tally for a pair of predictors on the occasion that they first appear in the same non-null group. At the end of the sampling iterations, the posterior estimate of the probability that a predictor is associated, or that a pair of predictors interact, is simply the frequency with which that event occurred.

Typically, I run *Sparse Partitioning* for a few hundred iterations; a few thousand for larger problems. This number may appear small compared to other MCMC based methods, which often run for tens or hundreds of thousands of iterations. However, it is necessary to bear in mind that *Sparse Partitioning* performs of the order N samplings within each iteration. I believe stating the number of iterations in this way is more representative, as the resolution of each posterior estimate will depend on how often the corresponding predictor is sampled, not the total number of samplings.

2.6.5 Discussion: Choice of Sampling Stages

Metropolis-Hastings theory greatly simplifies the task of designing an MCMC protocol. It allowed me to construct a proposal distribution however I pleased, then provided instructions for calculating an acceptance probability. As long as the resulting Markov Chain is irreducible, which in this case means that either every $\mathbb{G}$ or $\{\mathbb{G}, \boldsymbol{Z}\}$ pair is obtainable, its stationary distribution will match that of the posterior. Therefore, the goal became to pick proposal distributions which made the sampling as efficient as possible. In the pursuit of improved performance, I felt that were three ways that my method evolved from a more standard approach.

Perhaps most importantly, I allow the current partition to be altered in two different ways — either by changing an element of $\boldsymbol{I}$ or by changing an element of $\mathbb{G}$ — even though both ways on their own are in theory sufficient; any partition can be reached through Sampling Stage One changes while, if *Sparse Partitioning* also considered resampling empty group elements, the same would be true of Sampling Stage Two. However, suppose we wished to replace an associated predictor with one that is not associated. To achieve this switch requires changes to

two values of $\boldsymbol{I}$, and so would take at least two steps if only using the first sampling method. This severely reduces the chance of such a move happening, especially if the move must pass through a low scoring model. The second sampling method corrects this problem, as the move can be achieved by changing just one element of $\mathbb{G}$. Similarly, if only Sampling Stage Two type moves were permitted, for a predictor to switch between non-empty groups, at least two steps would be required.

Secondly, I tried to sample variables in an ordered manner so that, as far as possible, all variables were considered an equal number of times. Often MCMC based methods, for example, the MCMC version of Logic Regression (Kooperberg and Ruczinski, 2005), pick which variable to sample with replacement. The danger in high-dimensional problems is that such a system will result in some variables being repeatedly overlooked; very many iterations would be required to be confident that each variable has been sampled at least a few times. This explains the importance of describing each partition not just as a set of groups, $\mathbb{G}$, but also in terms of the indicator vector $\boldsymbol{I}$. While a far less concise definition, the length of $\boldsymbol{I}$ remains fixed, allowing sampling in a sequential fashion.

Based on this reasoning, it might then seem strange that I decided to sample only one component of $\mathbb{G}$ per iteration. In my implementation, partitions are stored in tree-like structures, each of which contains a group of associated predictors. Whenever a predictor is removed, the trees are reorganised to fill in the resulting gap. Because of this shuffling, it would be difficult to sample components of $\mathbb{G}$ in an ordered fashion. If I viewed this as a significant limitation, I'm sure a workable solution could be reached; however, as the number of choices for G_{kj} is necessarily no greater than $K \times S$, there is far less risk that a component of $\mathbb{G}$ is repeatedly overlooked. Secondly, I feel that the current set-up offers a reasonable balance; Sampling Stage One carries out the heavy-duty work, typically responsible for major changes in the model fit; by contrast, Sampling Stage Two performs fine-tuning, seeing whether the model can be tweaked by swapping in an unused predictor.

Finally, I prefer informative proposal distributions over uninformative ones, for which reason I chose Gibbs' sampling over Metropolis-Hastings where possible. This ties in with my preference for ordered sampling. Suppose *Sparse Partitioning* instead used an uninformative proposal distribution for G_{kj}, picking a new value purely at random from those possible. Each sampling would need to score only one additional partition, so this approach could perform approximately N repetitions in the time it takes to sample once from the conditional posterior of G_{kj}. Even so, when N is large, there is likely to be a large imbalance between the number of times each predictor is considered. Gibbs' sampling avoids this situation, as each predictor is considered for proposal once per iteration.

2.6.6 Discussion: Fixed or Variable p_g

At this point, I return to the discussion of whether or not to treat p_g as a variable. This has been frequently suggested to me as a possible change, as it gives the appearance of greater flexibility. Furthermore, p_g has a direct interpretation; it is the probability that predictor g is associated. Therefore, it would seem sensible to base posterior estimates that predictor g is associated on its posterior distribution $\mathbb{P}(p_g|\boldsymbol{X}, \boldsymbol{Y})$, rather than the current indirect method of counting how often the predictor features in the current partition.

Intuitively, allowing p_g to vary has always appeared unnecessary to me. Consider that we will be introducing N extra variables, the same number used to define a partition, so essentially doubling the parameter space. There would have to be a great benefit to justify increasing the complexity to such an extent. Secondly, introducing a variable p_g would not appear to add much extra depth. Each p_g would be implicitly linked to each I_g. A hierarchical set-up of this nature makes sense when each p_g relates to a number of other variables — a set-up I utilise when considering multivariate responses — but not when there is a one-to-one relationship. Consider the conditional posterior distribution of $\boldsymbol{p} = \{p_1, p_2, \ldots, p_N\}$:

$$\begin{aligned}\mathbb{P}(\boldsymbol{p}|\boldsymbol{X}, \boldsymbol{Y}, \mathbb{G}) &\propto \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{p}) \times \mathbb{P}(\boldsymbol{p}|\boldsymbol{X}, \mathbb{G}) \\ &\propto \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}) \times \mathbb{P}(\mathbb{G}|\boldsymbol{p}) \times \mathbb{P}(\boldsymbol{p}).\end{aligned}$$

Therefore

$$\mathbb{P}(p_g|p_{-g}, \boldsymbol{X}, \boldsymbol{Y}, \mathbb{G}) \propto p_g^{\mathbf{1}(I_g \neq 0)}(1 - p_g)^{\mathbf{1}(I_g = 0)} \times \mathbb{P}(p_g),$$

demonstrating that p_g's one-to-one relationship with I_g carries through to its posterior.

Being conjugate, the most obvious prior choice for p_g is the beta distribution $\beta(a_g, b_g)$, with suitable shape parameters a_g and b_g. The expected value of this distribution is $a_g/(a_g + b_g)$ so, for example, if $a_g = 1$, then $b_g = (1 - \bar{p}_g)/\bar{p}_g$, where $\bar{p}_g$ is the prior mean. Suppose *Sparse Partitioning* took this approach and instead sought the posterior mean of p_g. The first thing to notice is the conditional posterior distribution of p_g is either $\beta(a_g + 1, b_g)$ or $\beta(a_g, b_g + 1)$, depending on whether or not predictor g is declared associated by the current partition. Therefore, p_g's posterior mean necessarily lies within the interval

$$\left[\frac{a_g}{a_g + b_g + 1}, \frac{a_g + 1}{a_g + b_g + 1}\right].$$

When $a_g = 1$, these bounds are $1/(b_g + 2)$ and $2/(b_g + 2)$. While the form of the posterior distribution is not usually a valid criterion upon which to base a prior, we can see that posterior estimates based on p_g would be far less pleasing than those based on I_g, as the latter are only bounded by 0 and 1. Furthermore, it turns out that the Monte Carlo estimate of the posterior

mean of p_g will be a simple function of the marginal posterior probability estimate obtained by *Sparse Partitioning* when $\boldsymbol{p}$ is fixed.

$$\begin{aligned}\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}) \times \mathbb{P}(\boldsymbol{p}|\boldsymbol{X},\boldsymbol{Y}) &= \mathbb{P}(\boldsymbol{p}|\boldsymbol{X}) \times \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{p}) \\ &= \mathbb{P}(\boldsymbol{p}) \int_{\mathbb{G}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G},\boldsymbol{p}) \times \mathbb{P}(\mathbb{G}|\boldsymbol{X},\boldsymbol{p}) \, \mathrm{d}\mathbb{G} \\ &= \mathbb{P}(\boldsymbol{p}) \int_{\mathbb{G}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G}|\boldsymbol{p}) \, \mathrm{d}\mathbb{G}.\end{aligned}$$

We are interested in $\mathbb{E}(p_g|\boldsymbol{X},\boldsymbol{Y}) = \int_{\boldsymbol{p}} p_g \times \mathbb{P}(\boldsymbol{p}|\boldsymbol{X},\boldsymbol{Y}) \, \mathrm{d}\boldsymbol{p}$. Therefore, if we first multiply each side by p_g, then integrate with respect to $\boldsymbol{p}$, we obtain

$$\begin{aligned}&\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}) \times \mathbb{E}(p_g|\boldsymbol{X},\boldsymbol{Y}) \\ &\quad= \int_{\boldsymbol{p}} p_g \, \mathbb{P}(\boldsymbol{p}) \int_{\mathbb{G}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G}|\boldsymbol{p}) \, \mathrm{d}\mathbb{G} \, \mathrm{d}\boldsymbol{p} \\ &\quad= \int_{\mathbb{G}} \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G})}{B(\mathbb{G})} \int_{\boldsymbol{p}} p_g \prod_j \frac{p_j^{(a_j-1)}(1-p_j)^{(b_j-1)}}{\mathbb{B}(a_j,b_j)} p_j^{\mathbf{1}(I_j\neq 0)}(1-p_j)^{\mathbf{1}(I_j=0)} \, \mathrm{d}\boldsymbol{p} \, \mathrm{d}\mathbb{G} \\ &\quad= \int_{\mathbb{G}} \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G})}{B(\mathbb{G})} \int_{\boldsymbol{p}} p_g \prod_j \frac{p_j^{(a_j-1+\mathbf{1}(I_j\neq 0))}(1-p_j)^{(b_j-1+\mathbf{1}(I_j=0))}}{\mathbb{B}(a_j,b_j)} \, \mathrm{d}\boldsymbol{p} \, \mathrm{d}\mathbb{G},\end{aligned}$$

where $\mathbb{B}(a,b)$ represents the Beta function, the normalising constant of the beta distribution. By observing that

$$\begin{aligned}\int_{p_j} \frac{p_j^{(a_j-1+\mathbf{1}(I_j\neq 0))}(1-p_j)^{(b_j-1+\mathbf{1}(I_j=0))}}{\mathbb{B}(a_j,b_j)} &= \frac{\mathbb{B}(a_j+\mathbf{1}(I_j\neq 0), b_j+\mathbf{1}(I_j=0))}{\mathbb{B}(a_j,b_j)} \\ &= \frac{a_j^{\mathbf{1}(I_j\neq 0)} b_j^{\mathbf{1}(I_j=0)}}{a_j+b_j},\end{aligned}$$

and similarly

$$\int_{p_g} p_g \frac{p_g^{(a_g-1+\mathbf{1}(I_g\neq 0))}(1-p_g)^{(b_g-1+\mathbf{1}(I_g=0))}}{\mathbb{B}(a_g,b_g)} = \frac{a_g(a_g+1)^{\mathbf{1}(I_g\neq 0)} b_g^{\mathbf{1}(I_g=0)}}{(a_g+b_g)(a_g+b_g+1)},$$

we obtain

$$\begin{aligned}\mathbb{E}(p_g|\boldsymbol{X},\boldsymbol{Y}) &= \int_{\mathbb{G}} \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G})}{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})B(\mathbb{G})} \times \frac{a_g(a_g+1)^{\mathbf{1}(I_g\neq 0)} b_g^{\mathbf{1}(I_g=0)}}{(a_g+b_g)(a_g+b_g+1)} \prod_{j\neq g} \frac{a_j^{\mathbf{1}(I_j\neq 0)} b_j^{\mathbf{1}(I_j=0)}}{a_j+b_j} \, \mathrm{d}\mathbb{G} \\ &= \int_{\mathbb{G}} \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G})}{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X})B(\mathbb{G})} \times \frac{a_g}{a_g+b_g+1} \left(\frac{a_g+1}{a_g}\right)^{\mathbf{1}(I_g\neq 0)} \prod_j \frac{a_j^{\mathbf{1}(I_j\neq 0)} b_j^{\mathbf{1}(I_j=0)}}{a_j+b_j} \, \mathrm{d}\mathbb{G}.\end{aligned}$$

Consider what happens when *Sparse Partitioning* is run with $\boldsymbol{p}$ constant. Each p_g is fixed to

its prior mean value $a_g/(a_g + b_g)$, so

$$\begin{aligned}\mathbb{P}(\mathbb{G}) &= \prod_j \left(p_j^{\mathbf{1}(I_j \neq 0)} (1 - p_j)^{\mathbf{1}(I_j = 0)} \right) / B(\mathbb{G}) \\ &= \prod_j \left(\frac{a_j^{\mathbf{1}(I_j \neq 0)} b_j^{\mathbf{1}(I_j = 0)}}{a_j + b_j} \right) / B(\mathbb{G}).\end{aligned}$$

Therefore, the posterior mean of p_g can be written as

$$\begin{aligned}\mathbb{E}(p_g | \boldsymbol{X}, \boldsymbol{Y}) &= \int_{\mathbb{G}} \frac{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})}{\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}) B(\mathbb{G})} \times \frac{a_g}{a_g + b_g + 1} \left(\frac{a_g + 1}{a_g} \right)^{\mathbf{1}(I_g \neq 0)} \times \frac{\mathbb{P}(\mathbb{G})}{B(\mathbb{G})} \\ &= \int_{\mathbb{G}} \mathbb{P}(\mathbb{G}|\boldsymbol{X}, \boldsymbol{Y}) \times \frac{a_g}{a_g + b_g + 1} \left(\frac{a_g + 1}{a_g} \right)^{\mathbf{1}(I_g \neq 0)} \\ &= \int_{\mathbb{G}: I_g = 0} \mathbb{P}(\mathbb{G}|\boldsymbol{X}, \boldsymbol{Y}) \times \frac{a_g}{a_g + b_g + 1} + \int_{\mathbb{G}: I_g \neq 0} \mathbb{P}(\mathbb{G}|\boldsymbol{X}, \boldsymbol{Y}) \times \frac{a_g + 1}{a_g + b_g + 1}.\end{aligned}$$

By counting the number of times predictor g appears in the current partition, *Sparse Partitioning*'s posterior estimate of $\mathbb{P}(I_g \neq 0)$ represents a Monte Carlo estimate of

$$\int_{\mathbb{G}: I_g \neq 0} \mathbb{P}(\mathbb{G}|\boldsymbol{X}, \boldsymbol{Y}) \, \mathrm{d}\mathbb{G}.$$

Denoting this value by P, the Monte Carlo estimate of $\mathbb{E}(p_g|\boldsymbol{X}, \boldsymbol{Y})$ will be

$$(1 - P) \frac{a_g}{a_g + b_g + 1} + P \frac{a_g + 1}{a_g + b_g + 1},$$

which, as seems sensible, divides the domain of the posterior mean according to the fraction P.

Therefore, if it was considered appropriate to use a beta distribution prior for each p_g, instead of sampling its value at each step, the most efficient strategy to estimate the posterior mean of p_g would be to obey the method of *Sparse Partitioning*, then simply adjust the method's final results. Alternatively, the results of *Sparse Partitioning* could be viewed as the limiting case when $a_g(1 - \bar{p}_g)/\bar{p}_g = b_g \to 0$, whereby the prior distribution tends towards two point masses at 0 and 1, with weights $1 - \bar{p}_g$ and $\bar{p}_g$, respectively.

2.7 Simulation Study

In the next chapter, I continue the description of *Sparse Partitioning* by explaining how the method is designed to cope with issues that arise when analysing non-idealised dataset. However, at this point, I have provided sufficient details to create a working version of the method, so take the opportunity to demonstrate *Sparse Partitioning*'s potential using a simple set of

simulated datasets.

In total, I have performed ten simulation studies, with the aim of thoroughly testing *Sparse Partitioning* across a full range of scenarios. The complete results from all studies are provided in Chapter 4. For Study One, I considered datasets containing 100 samples, each typed for 1000 binary predictors. I examined three different underlying relationships, each involving three causal predictors. This study used perfect data; for example, there were no missing values, all causal predictors were observed and the predictors were uncorrelated. It formed the template for all subsequent studies, each of which then tested the effects of deviations from this idealised set-up. While Study One is far from realistic, during the development of *Sparse Partitioning*, I frequently used testing of this type as a sounding board to gauge whether progress was in the desired direction.

I picked the three underlying relationships in order to examine three contrasting models: one additive, one with a multiplicative interaction and one with a general interaction. These models are outlined in the following table:

Model	Underlying Relationship
I	$Y = X_1 + 1.5X_2 - 2X_3$
II	$Y = 1.5X_1 \times X_2 + X_3$
III	$Y = f(X_1, X_2) + X_3$, where $f(0,0) = 0$, $f(1,0) = 1$, $f(0,1) = 2$, $f(1,1) = -1$

Figure 2.4 compares the performance of *Sparse Partitioning* to seven of the existing methods outlined in the introduction: *Single* and *Pairs* (my implementations of basic one and two-predictors-at-a-time analyses); as well as *CART* (Classification and Regression Trees), *RF* (Random Forests), *SSS* (Shotgun Stochastic Search), *Logic* (Logic Regression) and *MARS* (Multivariate Adaptive Regression Splines). I have found that the performance of different methods will be greatly influenced by the "causal predictor frequency" which, for a binary predictor, is the (sample-specific) percentage of time it takes the value 1. When the predictors are SNPs, this term corresponds to each SNP's minor allele frequency. Therefore, as well as varying the underlying relationship, I also considered five different causal predictor frequencies: 0.05, 0.1, 0.2, 0.4 and '?'. The latter case corresponds to drawing each predictor's frequency from $\mathbb{U}(0.05, 0.95)$, a uniform distribution on the interval $[0.05, 0.95]$.

For each of the 15 scenarios, 100 datasets were created and each method was asked to declare its top three associations. I discuss why I took this decision more fully later on. In brief, I considered it a fairer comparison as it avoided the need to pick a declaration threshold

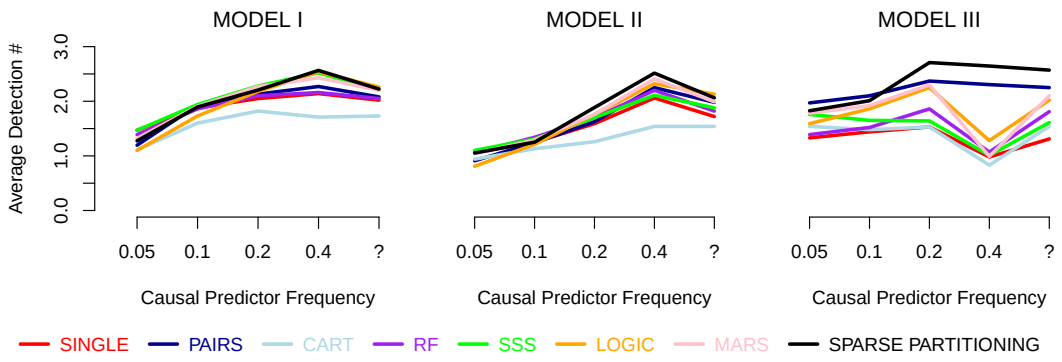


Figure 2.4: *Partial results of Simulation Study One. Each plot considers a different underlying relationship, which here are Models I, II and III (described in the main text). Within each plot, the lines report, for different causal predictor frequencies, the average number of causal predictors correctly detected by each method. The final frequency ('?') indicates that each causal predictor's frequency was drawn uniformly at random from the interval* $[0.05, 0.95]$.

or to plot the false discovery rate. Each line in Figure 2.4 plots the average number of causal predictors correctly declared by a particular method.

Sparse Partitioning, represented by the black line, is the best performing method under Model III. This is almost inevitable, as the general interaction contained in the simulated underlying relationship violates the assumptions of all other methods. However, it is reassuring that this success does not appear to come at the expense of performance under more simple models, as we see that *Sparse Partitioning* has also performed well in the other two scenarios. For Model I, the additive relationship, the method's line tracks very closely that of *SSS*, even though the latter method's underlying assumptions consider only additive models. Likewise, for the multiplicative relationship of Model II, *Sparse Partitioning* has matched the performance of *Logic*, whose underlying assumptions are tailored for relationships of this type.

These plots provided strong encouragement, and it was due to results of this nature that I formed, then cemented the view that it is better to risk overfitting the true model by being too general, than underfitting it by being too restrictive.

Chapter 3

Additional Features

The previous chapter describes core details of Sparse Partitioning*'s methodology, providing sufficient information with which to implement a working version. In this chapter, I explain additional features intended to cope with non-idealised datasets. I also consider issues of convergence and straightforward extensions of the method.*

3.1 Basic Preprocessing of Data

Having read in the data files, *Sparse Partitioning* performs some basic preprocessing steps designed to remove redundancies and standardise values. Firstly, the method searches for predictors where either all values are missing or all observed values are the same. In either case, these predictors are unable to offer evidence for an association, so are removed from the dataset and assigned posterior estimates of zero. If desired, the user can increase the level of filtering and, for example, require that each predictor has no more than 25% missing values and no fewer than 5 occurrences of the least commonly observed state. In a similar manner, *Sparse Partitioning* also checks that the response is not trivial, nor has too many missing values. When the response is continuous, its observed values are standardised to have mean 0 and variance 1.

By default, *Sparse Partitioning* scans the predictor set for obvious duplications, comparing each predictor with its 100 neighbours on either side. The similarity of two vectors is measured by calculating r^2, the square of their correlation. If missing values are encountered when comparing two predictors, the samples these correspond to are ignored for the purpose of calculating their similarity. If any pair of predictors is found to be identical ($r^2 = 1$), only the predictor with fewest missing values is retained and assigned the higher of the two prior probabilities of association. At the end of the analysis, each predictor removed through this pruning is assigned the same posterior estimates as its retained duplicate (i.e. given the same marginal posterior probability of association, the same posterior probability of interaction,

and so on).

If *Sparse Partitioning* is applied to a dataset containing two identical predictors, this would lead to unnecessary computation and potentially have an undesirable effect on the MCMC sampling. The desire for parsimony will generally prevent more that one of the duplicates featuring in the current model, so any evidence that these predictors are associated will be divided between the two sets of posterior estimates. *Sparse Partitioning* allows the user to vary the number of neighbours considered and reduce the r^2 threshold, in which case highly correlated predictors will also be considered duplicates. This comes in useful later on when analysing association study datasets exhibiting strong levels of LD.

After preprocessing the data, *Sparse Partitioning* calls the method *Single*, which performs one-predictor-at-a-time tests within both a frequentist and Bayesian framework, outputting p-values and posterior probabilities for each non-trivial predictor. (*Single* is explained in more detail in Chapter 4.)

3.2 Missing Data

When some predictor or response values are missing, one solution is to exclude from analysis all samples with incomplete data. However, in high-dimensional regression problems, even a tiny fraction of missing data can result in an undesirably high number of samples being removed. A second approach is to use imputation techniques before analysis to provide *Sparse Partitioning* with a complete dataset. For times when imputation is undesirable or inappropriate, *Sparse Partitioning* accepts missing values and incorporates their uncertainty into the analysis.

3.2.1 Missing Predictors

I explain only for the case of an identity or logit link function (Cases 1 and 2). For a probit link function (Case 3), the formulae are identical except for the addition of $\boldsymbol{Z}$.

Let's first augment the set of predictor values as $\boldsymbol{X} = \{\boldsymbol{O}, \boldsymbol{U}\}$, where the observed and unobserved values are represented by $\boldsymbol{O}$ and $\boldsymbol{U}$, respectively. Although $\boldsymbol{O}$ and $\boldsymbol{U}$ are not matrices, but rather structured lists, I will refer to their elements as if they were. Therefore, X_{ig} is represented by exactly one of O_{ig} or U_{ig}, depending on whether or not sample i records a value for predictor g. The posterior distribution now includes $\boldsymbol{U}$ as a variable:

$$\begin{aligned}\mathbb{P}(\mathbb{G}, \boldsymbol{U}|\boldsymbol{O}, \boldsymbol{Y}) &\propto \mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, \boldsymbol{U}, \mathbb{G}) \times \mathbb{P}(\mathbb{G}, \boldsymbol{U}|\boldsymbol{O}) \\ &= \mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, \boldsymbol{U}, \mathbb{G}) \times \mathbb{P}(\boldsymbol{U}|\boldsymbol{O}) \times \mathbb{P}(\mathbb{G}).\end{aligned}$$

Sparse Partitioning employs a very simple prior for $\boldsymbol{U}$. Firstly, the method assumes independence between both predictors and samples: $\mathbb{P}(\boldsymbol{U}|\boldsymbol{O}) = \prod_g \prod_i \mathbb{P}(U_{ig}|\boldsymbol{O})$. Then, the prior for each unobserved value is set to equal the frequency of observed values for that predictor: $\mathbb{P}(U_{ig} = u|\boldsymbol{O}) = \mathbb{P}(U_{ig} = u|O_g) \propto \sum_i \mathbf{1}(O_{ig} = u)$, for $u = 0, 1, 2$. The filtering of predictors performed before analysis ensures we will not have the situation where all values are missing, nor where all observed values are the same.

As we are primarily interested in the posterior distribution of partitions, we would ideally like to integrate across $\boldsymbol{U}$, but it can be readily appreciated that this will not normally be possible. When considering $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, \mathbb{G}) = \int_{\boldsymbol{U}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, \boldsymbol{U}, \mathbb{G}) \times \mathbb{P}(\boldsymbol{U}|\boldsymbol{O})\ \mathrm{d}\boldsymbol{U}$, we need only concern ourselves with the set $\boldsymbol{U}^\dagger$, those missing values corresponding to predictors which are declared associated by the current partition $\mathbb{G}$; all others have no impact on the likelihood and can be integrated out immediately. However, even with this shortcut, there will be between $2^{|\boldsymbol{U}^\dagger|}$ and $3^{|\boldsymbol{U}^\dagger|}$ states to consider.

Therefore, *Sparse Partitioning* resamples each unobserved predictor value once per iteration. This is carried out using Gibbs' sampling, at each step proposing U_{ig} from its conditional posterior distribution: $\mathbb{Q}(U_{ig}) = \mathbb{P}(U_{ig}|U^{-ig}, \boldsymbol{Y}, \boldsymbol{O}, \mathbb{G})$, where U^{-ig} represents $\boldsymbol{U}$ with element U_{ig} removed. This distribution takes the form

$$\mathbb{Q}(U_{ig}) \propto \mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, U_{ig}, U^{-ig}, \mathbb{G}) \times \mathbb{P}(U_{ig}|\boldsymbol{O}).$$

When predictor g is not declared associated by the current partition, the marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, U_{ig}, U^{-ig}, \mathbb{G})$ will not depend on U_{ig}, so a new value U^*_{ig} can be sampled directly from its prior. When predictor g is declared associated, it is necessary to calculate $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{O}, U_{ig}, U^{-ig}, \mathbb{G})$ for $U_{ig} = 0, 1, 2$, which is readily obtained by scoring the current partition using the three possible values of U_{ig}.

All other sampling stages remain the same, except that, whenever it is necessary to score a partition, $\boldsymbol{X}$ now represents both the observed predictor values and the current values of the unobserved predictors.

By adding in the sampling of $\boldsymbol{U}$, each MCMC iteration produces a draw from the joint posterior $\mathbb{P}(\mathbb{G}, \boldsymbol{U}|\boldsymbol{O}, \boldsymbol{Y})$. However, $\boldsymbol{U}$ is considered a nuisance parameter, so only the partition component is recorded. It would be straightforward for *Sparse Partitioning* to use the $\boldsymbol{U}$ component to estimate the posterior distribution of each U_{ig}, however, I would be hesitant to make claims concerning its accuracy; the space of $\boldsymbol{U}$ is typically far larger than that of $\mathbb{G}$, while for the majority of U_{ig}, those which do not contribute to the response, the data will offer no insight into their distribution.

Discussion: Incorporating Imputation Software

If the predictors are correlated, *Sparse Partitioning*'s default strategy for sampling missing values has obvious shortcomings. The prior is almost completely uninformative, in particular making no use of the information which can be gleaned from each predictor's neighbours. For the case of association study data, software packages exist for estimating missing genotypes based on the observed patterns of linkage disequilibrium (e.g. Clark, 1990; Stephens *et al.*, 2001; Howie *et al.*, 2009). These algorithms incorporate aspects of coalescent theory to provide much more accurate estimates of $\mathbb{P}(\boldsymbol{U}|\boldsymbol{O})$, with some able to incorporate reference genomes as well.

Armed with the results of imputation, the user has two options for each unobserved predictor: either they can accept the algorithm's best estimate and replace the missing value with this state; or, if the software outputs confidence intervals or posterior probabilities, they can use these as the basis for a prior distribution and override *Sparse Partitioning*'s default choice. Although it would seem sensible to always opt for the latter, the user should bear in mind that *Sparse Partitioning* will run faster and converge more assuredly when fewer values are missing. Therefore, when the confidence is high, they might prefer to err towards accepting the imputed values.

3.2.2 Missing Responses

Similar to how we handled missing predictors, let's augment the response values as $\boldsymbol{Y} = (Y_O, Y_U)^T$. Typically, the sets O and U correspond to observed and unobserved values, but this need not be the case. The distribution of the response values has been established by the regression equation, which provides $\mathbb{P}(Y_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}, \sigma^2)$ or $\mathbb{P}(Y_i|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta})$. Suppose Y_U represents the unobserved response values and consider the enlarged posterior distribution $\mathbb{P}(Y_U, \mathbb{G}|\boldsymbol{X}, Y_O)$ which treats each of these as a variable. Because each Y_i is considered an independent draw, we obtain

$$\begin{aligned}\mathbb{P}(\mathbb{G}, Y_U|\boldsymbol{X}, Y_O) = \mathbb{P}(\mathbb{G}, Y_U|X_O, X_U, Y_O) &\propto \mathbb{P}(Y_U, Y_O|X_O, X_U, \mathbb{G}) \times \mathbb{P}(\mathbb{G}) \\ &= \mathbb{P}(Y_U|X_U, \mathbb{G}) \times \mathbb{P}(Y_O|X_O, \mathbb{G}) \times \mathbb{P}(\mathbb{G}),\end{aligned}$$

where X_O and X_U denote the predictor values corresponding to the observed and unobserved response values. If Y_U is considered a nuisance parameter, the most efficient strategy is to integrate across its value. This leaves us with $\mathbb{P}(\mathbb{G}|X_O, Y_O) \propto \mathbb{P}(Y_O|X_O, \mathbb{G}) \times \mathbb{P}(\mathbb{G})$, the same form as before. Notice that the predictors corresponding to the missing response values are of no importance. Adopting this strategy is equivalent to analysing the data, having first discarded samples whose response values have not been observed. *Sparse Partitioning* facilitates this process by allowing the user to input all samples, then performing this exclusion on their

behalf.

There are occasions, however, when we wish to infer Y_U, so *Sparse Partitioning* offers the option of predicting response values. Prediction need not be limited to estimating missing response values; for example, the "prediction" of observed values proves useful during cross-validation, described later on. When predicting Y_U, we seek the posterior predictive distribution, which takes the form $\mathbb{P}(Y_U|\boldsymbol{X}, Y_O)$. Typically, a method will consider prediction post-analysis, by applying the results of one run to a second set of predictor values. This relies on the method returning an exact estimate of the underlying relationship, in which case it is straightforward to apply the regression model to the new predictors. This is not possible with *Sparse Partitioning* for two reasons: $\boldsymbol{\Theta}$ is considered a nuisance parameter, so integrated out where possible; and the method estimates the distribution of $\mathbb{G}$, rather than finding a single best fitting value.

Sparse Partitioning uses two techniques for prediction, both of which rely on Monte Carlo integration: the first resamples the value of Y_U, while the second calculates its posterior predictive mean. For reasons I will explain later, the second strategy is preferable, but not always possible. Therefore, the approach taken by *Sparse Partitioning* depends on the situation and is most easily described by a table:

Case	**Response - Link**	**Situation**	**Action (Once Per Iteration)**
1	Continuous - Identity	All $X_{\boldsymbol{U}}$ Observed	Calculate $\mathbb{E}(Y_U\|\boldsymbol{X}, Y_O, \mathbb{G})$
		Some $X_{\boldsymbol{U}}$ Missing	Resample Y_U and Missing X_U
2	Binary - Logit	Any	Resample Y_U and Missing X_U
3	Binary - Probit	All $X_{\boldsymbol{U}}$ Observed	Calculate $\mathbb{E}(Z_U\|\boldsymbol{X}, Y_O, \mathbb{G})$
		Some $X_{\boldsymbol{U}}$ Missing	Resample Y_U and Missing X_U

Resampling of Y_U

When this strategy is chosen, each element of Y_U is resampled once per iteration. If any elements of X_U are unobserved, they will be resampled at the same time as any other missing predictors. Suppose we wish to resample $Y_i \in Y_U$. When the response is continuous, the conditional posterior distribution of Y_i is proportional to the marginal likelihood:

$$\begin{aligned}\mathbb{P}(Y_i|Y_{-i}, \boldsymbol{X}, \mathbb{G}) &\propto \mathbb{P}(Y_i, Y_{-i}|\boldsymbol{X}, \mathbb{G}) \\ &\propto (\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})^{-\frac{n}{2}} \\ &= (\boldsymbol{Y}(\boldsymbol{I}_n - \boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^T)\boldsymbol{Y})^{-\frac{n}{2}},\end{aligned}$$

which is not of recognisable form. Therefore, *Sparse Partitioning* proposes from a standard normal distribution $\mathbb{Q}(Y_i) = \phi(Y_i)$ and accepts the new value Y_i^* with probability $\min(1, \alpha)$,

where

$$\alpha = \frac{\mathbb{P}(Y_i^*, Y_{-i}|\boldsymbol{X}, \mathbb{G})}{\mathbb{P}(Y_i, Y_{-i}|\boldsymbol{X}, \mathbb{G})} \times \frac{\phi(Y_i)}{\phi(Y_i^*)}.$$

The first fraction is equal to the ratio of the partition scores calculated using the proposed and current value of Y_i. Here, it is useful that the response values are standardised before analysis, as it ensures the proposal distribution has appropriate shape and scale.

When the response is binary and a logit link function is used, the conditional posterior distribution for a missing value can be found explicitly by calculating the probability it takes 0 or 1. This is essentially what *Sparse Partitioning* does, always proposing to toggle the response, $\mathbb{Q}(Y_i^*|Y_i) = \delta_{\{1-Y_i\}}$, then accepting with probability $\min(1, \alpha)$, where

$$\alpha = \frac{\mathbb{P}(Y_i^*, Y_{-i}|\boldsymbol{X}, \mathbb{G})}{\mathbb{P}(Y_i, Y_{-i}|\boldsymbol{X}, \mathbb{G})} \times \frac{\mathbb{Q}(Y_i|Y_i^*)}{\mathbb{Q}(Y_i^*|Y_i)}.$$

The second fraction has value one, as both proposals are the only moves permitted. Therefore, α equals the ratio of the partition scores calculated using the proposed and current values of Y_i.

Now, consider the case of a binary response and probit link function. Bearing in mind that Y_i determines the sign of Z_i, if we sampled their values independently, neither the values of $\boldsymbol{Y}$ nor the signs of $\boldsymbol{Z}$ would change. Therefore, when Z_i corresponds to a response which is missing, *Sparse Partitioning* amends its sampling to appreciate that its sign is no longer fixed. Instead of proposing a new value from a folded standard normal, a standard normal is used: $\mathbb{Q}(Z_i) = \phi(Z_i)$. The probability of accepting the proposed value Z_i^* remains equal to $\min(1, \alpha)$, where

$$\alpha = \frac{\mathbb{P}(\mathbb{G}, Z_i^*, Z_{-i}|\boldsymbol{X}, \boldsymbol{Y})}{\mathbb{P}(\mathbb{G}, Z_i, Z_{-i}|\boldsymbol{X}, \boldsymbol{Y})} \times \frac{\phi(Z_i)}{\phi(Z_i^*)} = \frac{\mathbb{P}(Z_i^*, Z_{-i}|\boldsymbol{X}, \mathbb{G}) \times \mathbb{P}(\mathbb{G})}{\mathbb{P}(Z_i, Z_{-i}|\boldsymbol{X}, \mathbb{G}) \times \mathbb{P}(\mathbb{G})} \times \frac{\phi(Z_i)}{\phi(Z_i^*)},$$

and once again, the first fraction equals the ratio of the partition scores calculated using the proposed and current values of Z_i.

Calculation of Posterior Predictive Mean

This strategy does not allow for missing elements of X_U, so it is suitable only when all predictors corresponding to Y_U have been observed. It also requires that the link function is the identity or probit function (Cases 1 or 3), for reasons that will become clear shortly. First, I explain for the identity, then mention the slight change when instead a probit link function is used.

To obtain the full posterior predictive distribution, it would be necessary to integrate the raw likelihood of Y_U across $\mathbb{G}$, $\boldsymbol{\Theta}$ and σ^2, according to these parameters' posterior distribu-

tion. As summation over partitions is not possible, *Sparse Partitioning*'s solution is to pick a statistic, in this case the posterior predictive mean, and estimate its value using Monte Carlo integration. To do this, first consider $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \sigma^2) = \mathbb{P}(Y_O, Y_U|\boldsymbol{X}, \mathbb{G}, \sigma^2)$, the marginal likelihood given the current variance. Using earlier calculations, we can obtain

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \sigma^2) = r^{\frac{D}{2}}|\boldsymbol{B}|^{-\frac{1}{2}}(2\pi\sigma^2)^{-\frac{n}{2}} \times \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y}^T(\boldsymbol{I}_n - \boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^T)\boldsymbol{Y})\right\},$$

which has the form of a multivariate normal distribution with (inverse) variance matrix $\sigma^{-2}\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma} = \boldsymbol{I}_n - \boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^T$. The symmetric matrix $\boldsymbol{\Sigma}$ can be partitioned as

$$\boldsymbol{\Sigma} = \begin{vmatrix} \Sigma_{UU} & \Sigma_{UO} \\ \Sigma_{OU} & \Sigma_{OO} \end{vmatrix},$$

where $\Sigma_{OU} = (\Sigma_{UO})^T$. Here, Σ_{UU} and Σ_{OO} correspond to the variances of Y_U and Y_O, respectively, while Σ_{UO} corresponds to their covariance. This tells us that $\mathbb{P}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G}, \sigma^2)$, the conditional posterior predictive distribution given the current variance, is also multivariate normal, with mean vector $-(\Sigma_{UU})^{-1}\Sigma_{UO}Y_O$ and (inverse) variance matrix $\sigma^{-2}\Sigma_{UU}$. Notice that this mean vector does not depend on σ^2, so will also equal the mean of $\mathbb{P}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G})$ and there is no need to integrate with respect to σ^2. Therefore, at each iteration, *Sparse Partitioning* calculates and records $\mathbb{E}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G})$, and these values are, at the end, averaged over to obtain an estimate of the posterior predictive mean of Y_U.

When a probit link function is used, the conditional posterior predictive distribution $\mathbb{P}(Z_U|\boldsymbol{X}, Y_O, Z_O, \mathbb{G}) = \mathbb{P}(Z_U|\boldsymbol{X}, Z_O, \mathbb{G})$ is equal to its continuous response counterpart with Y_U and Y_O replaced by Z_O and Z_U, and σ^2 set to 1. The removal of σ^2 means we can obtain this distribution explicitly, so are not limited to considering only its mean. Nonetheless, I persist with calculating its expected value for each iteration, which later leads to an estimate of $\mathbb{E}(Z_U|\boldsymbol{X}, Y_O)$. Rather than return predicted values for Z_U, each component is converted into a probability that the corresponding response value equals 1, using the transformation $a \to \Phi(a)$.

To adopt this strategy with a logit link function would require calculation of $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$, which can only be done when values for all variables are provided. In theory, it would be possible to calculate the conditional posterior predictive distribution $\mathbb{P}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G})$, by evaluating the marginal likelihood for each of the $2^{|U|}$ possible values of Y_U. Whether this was possible in practice would depend on the number of response values unobserved, so for consistency *Sparse Partitioning* opts against this strategy.

As well as allowing for missing elements of X_U, the first strategy has the advantage of providing a sampling from the posterior predictive distribution of Y_U, so the user is free to

examine any property they please. For example, they might wish to check the predicted variance is in line with that observed. Such assessment is not easily possible for the second strategy, at least for the case of a continuous response, as only estimates of statistics of the posterior predictive distribution, rather than samplings from, are obtained. However, in all other respects, I consider the second strategy preferable. As I discussed, there is no reason to involve the auxiliary variables X_U and Y_U when estimating $\mathbb{P}(\mathbb{G}|X_O, Y_O)$, but doing so will likely impede convergence. I also prefer the way the second strategy considers the joint distribution of missing response values, rather than considering each marginal distribution in turn. Furthermore, the prediction accuracy of the first strategy depends on the convergence of draws from $\mathbb{P}(\mathbb{G}, Y_U|X_U, X_O, Y_O)$, while in the second it depends only on the convergence of draws from $\mathbb{P}(\mathbb{G}|X_O, Y_O)$. For all these reasons, when calculation of $\mathbb{E}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G})$ is possible, the second strategy takes priority.

3.3 Confounding

When I talk about confounding variables, I refer to any factors which affect the underlying relationship, but in whose involvement we are not directly interested. For example, we might wish to analyse a dataset where a sample's gender plays a major role in determining its response value. If we are interested in studying the effect of sex, we could treat it just like any other predictor, in which case we could ask to what extent it is involved, whether it interacts with other predictors and so on. If we are not interested in these questions, but merely consider sex a nuisance variable, then it would be classed as confounding.

Quite often, confounding arises from variables which are difficult to observe. In association studies, one such example is relatedness. It will be very unlikely the experiment obtains exact pedigree information for all samples. I later analyse data obtained from the plant *Arabidopsis thaliana*. Here, more than in humans, relatedness is a major concern as accessions are often picked from heavily inbred lines. Suppose, in an experiment, two samples are so closely related that their values are almost identical. This will produce a "pseudo-duplicate". If one of these samples provides evidence for a particular association, then the second sample will likely magnify this evidence, whether or not this support is warranted. As these two samples are highly related, they will likely have developed in similar environments relative to the rest of the sample. We might consider this a more plausible reason for their similar responses, as opposed to a genetic explanation. Therefore, if we have pedigree information for the samples, it is prudent to incorporate this in the analysis. Even when this information is not available, methods exist for its estimation (discussed in Astle and Balding, 2009).

Let the columns of the matrices $\boldsymbol{\Psi}$ and $\boldsymbol{\Omega}$ represent confounding variables. These variables

are divided according to the way *Sparse Partitioning* allows them to affect the underlying relationship; variables contained in $\boldsymbol{\Psi}$ are allowed to interact with the standard predictors $\boldsymbol{X}$, those in $\boldsymbol{\Omega}$ are assumed to contribute independently and additively. The regression equation becomes $l(\mathbb{E}(\boldsymbol{Y})) = f(\boldsymbol{X}, \boldsymbol{\Psi}) + \boldsymbol{\Omega}\,\boldsymbol{\omega}$, where $\boldsymbol{\omega}$ contains the regression coefficients corresponding to variables in $\boldsymbol{\Omega}$.

As *Sparse Partitioning* is only designed to consider interactions between tertiary predictors, the method requires that all variables of $\boldsymbol{\Psi}$ are coded as such. Consequently, the method is unable to consider interactions with quantitative confounding variables. I discuss the impact of this limitation after explaining how *Sparse Partitioning* tries to correct for confounding.

Sparse Partitioning automatically accounts for the effect of variables in $\boldsymbol{\Omega}$, assigning $\boldsymbol{\omega}$ a normal prior distribution with mean 0 and variance σ^2/r' or $1/r'$, depending on whether the response is continuous or binary. Consider the effect on the marginal likelihood when the response is continuous (Case 1):

$$\begin{aligned}
\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}) &= \int_{\boldsymbol{\Theta},\sigma^2} \int_{\boldsymbol{\omega}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Theta}, \sigma^2, \boldsymbol{\omega}) \times \mathbb{P}(\boldsymbol{\omega})\ \mathrm{d}\boldsymbol{\omega} \times \mathbb{P}(\boldsymbol{\Theta}, \sigma^2)\ \mathrm{d}\boldsymbol{\Theta}\ \mathrm{d}\boldsymbol{\sigma}^2 \\
&= \int_{\boldsymbol{\Theta},\sigma^2} \int_{\boldsymbol{\omega}} (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{J\Theta} - \boldsymbol{\Omega\omega})^T(\boldsymbol{Y} - \boldsymbol{J\Theta} - \boldsymbol{\Omega\omega})\right\} \\
&\quad \times (2\pi\sigma^2/r')^{-\frac{D'}{2}} \exp\left\{-\tfrac{r'}{2\sigma^2}\boldsymbol{\omega}^T\boldsymbol{\omega}\right\}\ \mathrm{d}\boldsymbol{\omega} \times \mathbb{P}(\boldsymbol{\Theta}, \sigma^2)\ \mathrm{d}\boldsymbol{\Theta}\ \mathrm{d}\boldsymbol{\sigma}^2,
\end{aligned}$$

where D' denotes the number of variables in $\boldsymbol{\Omega}$. Using similar steps to when we earlier integrated over $\boldsymbol{\Theta}$, we obtain

$$\begin{aligned}
\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}) &= \int_{\boldsymbol{\Theta},\sigma^2} r'^{\frac{D'}{2}} |\boldsymbol{B}'|^{-\frac{1}{2}} (2\pi\sigma^2)^{-\frac{n}{2}} \times \exp\left\{-\tfrac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{J\Theta})^T\boldsymbol{C}(\boldsymbol{Y} - \boldsymbol{J\Theta})\right\} \\
&\quad \times \mathbb{P}(\boldsymbol{\Theta}, \sigma^2)\ \mathrm{d}\boldsymbol{\Theta}\ \mathrm{d}\boldsymbol{\sigma}^2,
\end{aligned}$$

where $\boldsymbol{C} = \boldsymbol{I}_n - \boldsymbol{\Omega}\boldsymbol{B}'^{-1}\boldsymbol{\Omega}^T$, with $\boldsymbol{B}' = \boldsymbol{\Omega}^T\boldsymbol{\Omega} + r'\boldsymbol{I}_{D'}$. When the response is binary and a probit link function is used (Case 3), $\boldsymbol{Y}$ is replaced by $\boldsymbol{Z}$ and σ^2 fixed at 1, but otherwise the effect on the marginal likelihood is identical. With this small adjustment, *Sparse Partitioning* is able to continue as before.

We see that the introduction of $\boldsymbol{\Omega}$ is equivalent to ignoring its presence, but altering the likelihood assumption to suppose that $\boldsymbol{Y}$ is now drawn from a multivariate normal distribution with mean $\boldsymbol{J\Theta}$ and variance matrix $\sigma^2\boldsymbol{C}^{-1}$. Viewed the other way round, when confounding factors introduce correlations between response values, as clearly is the case with relatedness, the response values can equally be thought of as independent draws, but with a certain quantity added to their underlying relationship values.

Allowing for $\boldsymbol{\Omega}$ slows down computation, as the introduction of $\boldsymbol{C}$ (size $n \times n$) complicates existing calculations. However, the value of $\boldsymbol{C}$ remains constant throughout analysis and so, for example, $\boldsymbol{CY}$ only changes when missing response values are resampled. Tricks of this nature are able to offset the increase in computation time to some extent. Nonetheless, acknowledging $\boldsymbol{\Omega}$'s presence in every calculation will have a noticeable effect; for example, when the sample size is a few hundred, this will typically result in iterations taking about twice as long.

Fortunately, it is generally acceptable to adjust for the effect of $\boldsymbol{\Omega}$ before analysis, by replacing the response values with the residuals $\boldsymbol{Y} - \boldsymbol{\Omega}\hat{\boldsymbol{\omega}}$, where $\hat{\boldsymbol{\omega}}$ is a suitable estimate of $\boldsymbol{\omega}$. There are two justifications for this. Consider the linear model $\boldsymbol{Y} = \boldsymbol{J}\Theta + \boldsymbol{\Omega\omega}$ when all variables are orthogonal. If $< \cdot\,, \cdot >$ denotes the inner product of two vectors, we can calculate least squares estimates for each coefficient by "dotting" both sides with the corresponding variable:

$$\hat{\Theta}_j = \frac{< \boldsymbol{Y}, J_j >}{< J_j, J_j >} \quad \text{and} \quad \hat{\omega}_{j'} = \frac{< \boldsymbol{Y}, \Omega_{j'} >}{< \Omega_{j'}, \Omega_{j'} >}.$$

Therefore, for least squares regression, the overall model fit can be calculated in a stepwise fashion, at each step regressing the current residuals on the next predictor. A similar argument can be used in the Bayesian setting to justify first regressing $\boldsymbol{Y}$ on $\boldsymbol{\omega}$, then replacing $\boldsymbol{Y}$ with $\boldsymbol{Y} - \boldsymbol{\Omega}\hat{\boldsymbol{\omega}}$, where $\hat{\boldsymbol{\omega}}$ is the posterior mean from this first regression. In fact, we only require that each column of $\boldsymbol{\Omega}$ is perpendicular to each column of $\boldsymbol{J}$, the case when their observed values are uncorrelated. Over reasonable sample sizes, we might expect this to be so.

Should this reasoning not seem sound, a second explanation might be more readily accepted. By automatically considering the contribution to the model of variables in $\boldsymbol{\Omega}$, this, in effect, assigns a prior probability of 1 that they are associated. This is in stark contrast, in sparse problems at least, to the standard predictors which are assigned very small probabilities. This relative weighting is reasonable when the confounding variables are "major factors", as then we would consider it far more likely that they influence the response than any single standard predictor. If $\boldsymbol{X}$ and $\boldsymbol{\Omega}$ are considered together in the regression model, their corresponding regression coefficients will be competing over any variation both are able to explain. However, our prior belief heavily favours $\boldsymbol{\Omega}$ winning this battle convincingly, justifying why we might account for its effect beforehand.

When the response is binary and a logit link function is used (Case 2), the marginal

likelihood becomes

$$\begin{aligned}\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) &= \int_{\boldsymbol{\Theta},\boldsymbol{\omega}} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G},\boldsymbol{\Theta},\boldsymbol{\omega}) \times \mathbb{P}(\boldsymbol{\Theta}) \times \mathbb{P}(\boldsymbol{\omega})\ \mathrm{d}\boldsymbol{\Theta}\ \mathrm{d}\boldsymbol{\omega} \\ &= \int_{\boldsymbol{\Theta},\boldsymbol{\omega}} \prod_i [l^{-1}(\boldsymbol{J}\boldsymbol{\Theta}+\Omega\boldsymbol{\omega})]^{Y_i}\ [1-l^{-1}(\boldsymbol{J}\boldsymbol{\Theta}+\Omega\boldsymbol{\omega})]^{(1-Y_i)} \times \mathbb{P}(\boldsymbol{\Theta},\boldsymbol{\omega})\ \mathrm{d}\boldsymbol{\Theta}\ \mathrm{d}\boldsymbol{\omega} \\ &= \int_{\boldsymbol{\Theta}^+} \prod_i [l^{-1}(\boldsymbol{J}^+\boldsymbol{\Theta}^+)]^{Y_i}\ [1-l^{-1}(\boldsymbol{J}^+\boldsymbol{\Theta}^+)]^{(1-Y_i)} \times \mathbb{P}(\boldsymbol{\Theta}^+)\ \mathrm{d}\boldsymbol{\Theta}^+, \mathbb{P}(\mathbb{G}, Y_U|\boldsymbol{X}, Y_O)\end{aligned}$$

where $\boldsymbol{J}^+ = [\boldsymbol{J}\ \boldsymbol{\Omega}]$ and $\boldsymbol{\Theta}^+ = (\boldsymbol{\Theta}, \omega)$. A Laplace approximation can be used to integrate across this extended linear model. We would prefer to be able to integrate out $\boldsymbol{\omega}$ beforehand, as this integral is common to all models. However, this is not possible using a Laplace approximation, as it an analytic technique which requires numerical values for $\boldsymbol{\Theta}$.

This obstacle can have a dramatic effect on computation time. If we wish to include confounding factors based on sample relatedness, this typically produces an additional n covariates in the model, one for each sample. In general, this will greatly increase the degrees of freedom of the linear model, making the Newton-Raphson method much slower. Therefore, when $\boldsymbol{\Omega}$ is included in a problem with a binary response, I highly recommend using a probit link function or allowing the method to regress out the effect of $\boldsymbol{\Omega}$ in advance of analysis, which can be justified by an argument similar to that discussed for the continuous case.

The second set of confounding variables, those contained in $\boldsymbol{\Psi}$, are easy to include, as *Sparse Partitioning* simply treats them as additional columns of $\boldsymbol{X}$. The method will return posterior estimates for these variables, as it does for all standard predictors. These variables require a prior probability of association. As we generally consider confounding variables more likely to influence the response, these probabilities should typically be set higher than for the standard predictors. If we are certain of their involvement, they can be added to the list of predictors *Sparse Partitioning* must include.

If confounding variables are supplied, these are incorporated during the one-predictor-at-a-time tests performed by *Single* by altering its null and alternative hypotheses to include a contribution from $\boldsymbol{\Omega}$ and $\boldsymbol{\Psi}$.

Discussion: Assumption that $\boldsymbol{X}$ and $\boldsymbol{\Omega}$ do not interact

Sparse Partitioning is only set up to consider interactions between tertiary variables. In particular, this prevents it from exploring interactions involving quantitative confounding variables. Consider the case of population confounding, which refers to trends and correlations which appear across populations as a result of migration and geographic differences.

Consider an idealised scenario, where each individual is sampled from one of two distinct populations. In this simple case, only a single, two-state vector would be required to indicate each sample's population status and this could readily be included as an additional predictor in the model. In general the stratification will be far more complex. There will often be a number of loosely defined populations, with varying levels of overlap ("admixture") between each pair. To describe such a complex situation requires a few vectors where, say, each vector indicates the fraction of a sample originating from a particular population. Even when very detailed migration histories are available, it will not normally be possible to reconstruct these population vectors with much accuracy.

Fortunately, methods have been developed to estimate these population vectors from the sample dataset. For a single, idealised population, Hardy-Weinberg equilibrium dictates that a SNP's allele frequencies will remain constant throughout generations. In particular, when a SNP is biallelic, the frequencies of homozygous wildtype, heterozygous and homozygous mutant states should obey the ratio $p^2 : 2p(1-p) : (1-p)^2$. When populations are merged, unless a SNP's allele frequencies happen to be the same across all populations, this equilibrium will be destroyed. This principle is exploited by the method `STRUCTURE` (Pritchard *et al.*, 2000) which, loosely speaking, attempts to partition the sample so that within each group Hardy-Weinberg equilibrium is restored.

The extent to which a SNP obeys Hardy-Weinberg equilibrium will, in practice, depend on many factors. These include the validity of the theory's assumptions, which in particular suppose randomly mating individuals and the absence of selective forces. As a result, some SNPs will be more informative of population differences and these will contribute most to `STRUCTURE`'s estimates of the population vectors. The method `EIGENSTRAT` (Patterson *et al.*, 2006) takes an alternative approach, one based instead on principal component analysis. Here, the idea of detecting informative variants is more explicit, as the algorithm directly sets out to find data axes across which the differences between individuals' genetic data are highlighted.

Therefore, when considering how much *Sparse Partitioning* suffers for not being able to consider interactions with population covariates, it perhaps helps to bear in mind the biological interpretation of these interactions. Essentially, each population variable is a statistic corresponding to the group of genetic variants which best distinguish that population. It is much easier to conceive reasons why single variants might interact, than explain why groups of scattered variants might. Even so, *Sparse Partitioning*'s inability to consider interactions with quantitative confounding variables clearly affects its generality to some level, so this should be acknowledged.

3.4 Multicollinearity

This section considers the case when strong correlations exist between predictors. This is perhaps most relevant when analysing data from fine-scale association studies, where high levels of linkage disequilibrium are regularly observed. To demonstrate the effect that correlations can have on a regression method, I use as an example a 200 kbp region of Human Chromosome 1 centred on the `MTHFR` gene, the focus of one of the real datasets examined in the next chapter.

At the top of Figure 3.1 is an LD plot obtained from the latest data release of the HapMap Project (version 3, release 2; The International HapMap Consortium, 2007). This provides values of correlation squared (r^2) between pairs of SNPs, with darker squares indicating values closer to 1. Conserved sequences — regions which, relatively speaking, are affected less by recombination events within the population — will generally show up as distinct neighbourhoods of highly correlated SNP. The (faint) triangles represent HapMap's attempt to identify these "haploblocks". Note that, for the purpose of the diagram, the SNPs have been evenly spaced.

The middle plot shows p-values obtained when the method *Single* regresses expression levels of the `MTHFR` gene against SNP values. Although the dataset I used for this analysis comes from an earlier release, one containing fewer individuals and SNPs, there is still a marked similarity between the top two plots. There is a high concentration of strong associations within the haploblock immediately downstream of the gene, whose actual endpoints I have marked with a bold horizontal line. The natural assumption is that at most one SNP in the region (either observed or unobserved) is truly causal, so any evidence for other associations is likely spurious and a result of strong correlations with the causal SNP.

This example highlights one of the major drawbacks of one-predictor-at-a-time methods when applied to association studies. Even if their underlying relationship assumption is true and there actually is only one causal SNP, by performing independent tests they are unable to consider the effect of LD. They will find it hard to distinguish between a true association and a highly correlated false positive, and it will be left to the experimenter to make this call. To be able to appreciate the correlations present, a method must consider more than one predictor at once.

The bottom plot of Figure 3.1 presents results from a primitive attempt at multiple regression. It shows the regression coefficients which minimise the penalised least squares $(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\Theta})^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\Theta}) + \boldsymbol{\Theta}^T\boldsymbol{\Theta}$, a strategy matching that of ridge regression. Coefficients further from the y-axis suggest stronger evidence for association. Despite the crudeness of this method, the results clearly appear more worthwhile. The peaks no longer simply mirror

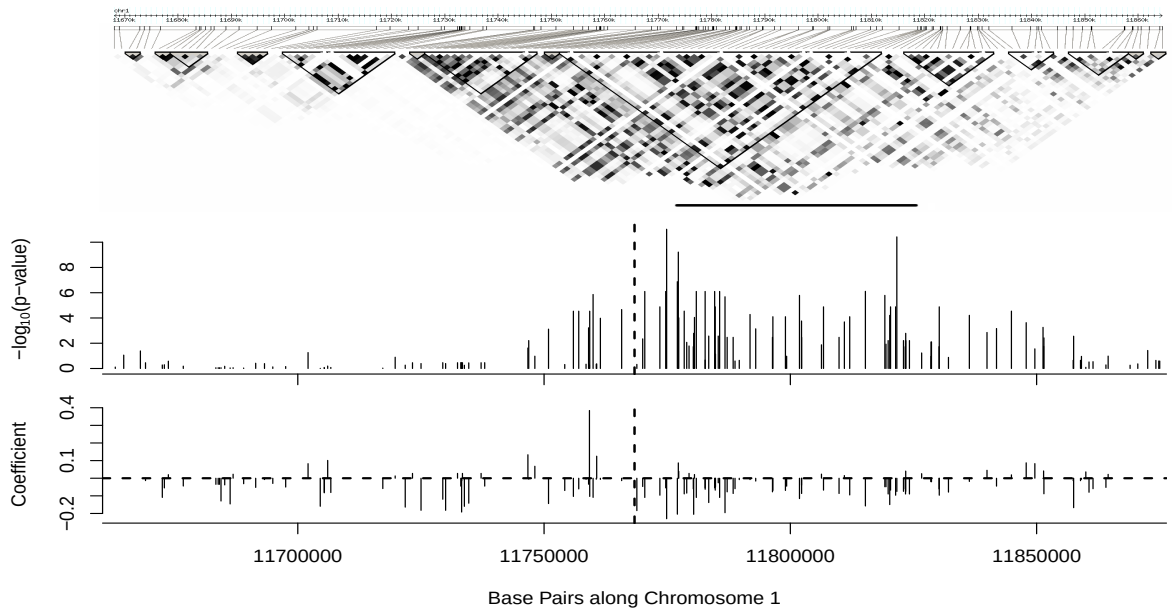


Figure 3.1: *Effects of linkage disequilibrium in association studies. Each plot refers to a 200 kbp region of Human Chromosome 1, centred on the* `MTHFR` *gene. The top plot provides pairwise correlation values between SNPs, obtained from 205 individuals from the HapMap project. The shade of each diamond indicates the extent of linkage disequilibrium observed between a pair of SNPs; darker shades reflect higher correlation. The middle plot presents p-values from the method* Single*, regressing the gene's expression on a selection of the HapMap SNPs. The most extreme p-values lie within, or very close to, a region of high LD identified by HapMap (marked by a horizontal line). The bottom plot displays regression coefficients from a primitive version of ridge regression. This method is able to consider the joint contribution of predictors, and so appreciates that many SNPs are highly correlated. The result is a notable change in the location of the strongest hits, identified as the SNPs with regression coefficients furthest from zero.*

the patterns of correlations. Among the group of correlated predictors competing to explain the variance, one has prevailed and been accounted for, leaving the remainder to fight over the residual. *Sparse Partitioning* operates in a similar fashion; by analysing all predictors at once, it is able to consider the combined way predictors contribute. With an underlying model more detailed than ridge regression, it should be able to do this more accurately.

Ridge regression is an example of a shrinkage method, as the penalty term is a continuous function of the regression coefficients. For these methods, inclusion and exclusion of predictors is judged on a continuous scale, corresponding to how much coefficients are pushed away from or towards zero. By contrast, variable subset selection methods operate in a discrete manner, insisting each predictor is either in or out of the model. These methods are generally affected more by correlated predictors. For example, suppose in an association study a causal SNP has not been observed, but displays very high LD with two observed markers. Of the two markers, the one with the highest sample correlation with the causal SNP will probably be declared associated. However, if these two sample correlations are very close, repeating the experiment with a slightly different set of samples might change which marker is declared associated.

For this reason, variable subset selection methods can display high variability across experiments. In this situation, the advantage of a shrinkage method is that the uncertainty should be reflected in the two predictors' coefficient values, making the experimenter aware of how close the sample came to producing a different result. *Sparse Partitioning* operates in the style of a subset selection manner. Its penalty term has a discrete component and at each step it decides whether each predictor is associated or not. This suggests it might share the same drawbacks. However, while this is true on a per iteration basis, its posterior estimates are based on averages. Therefore, provided the MCMC sampling achieves its goal of convergence, uncertainty should be reflected in the estimates.

3.4.1 Removing High Correlations

There are two main reasons for filtering predictors with identical observed values. The first is a computational consideration. Leaving duplicate predictors in the dataset will result in many unnecessary calculations and slow down convergence. The second reason concerns interpretability. If duplicate predictors correspond to an association, the posterior evidence for this will be divided among the copies. It will be quite possible that *Sparse Partitioning* has detected the signal from the association, but this is not visible because the posterior evidence has been spread out across a number of predictors.

Similar reasoning applies when there is a group of very highly correlated predictors, as this will lead to sets of partitions with almost identical fit to the data. Therefore, when mul-

ticollinearity is present, I strongly recommend utilising *Sparse Partitioning*'s option to first filter the predictor set. The choice of r^2 threshold provides a trade-off between precision and speed. Lowering the value will reduce the resolution of results and can only weaken the signal present in the data. However, the method will converge faster and more assuredly, and there will be less chance that the true signal is diluted across multiple predictors. I generally opt for $r^2 = 0.8$, which has been shown able to account for reasonable variation (W de Bakker *et al.*, 2005). Corresponding to this threshold, is a setting which dictates the size of neighbourhood the method should search for duplicates. This exists for convenience, as ideally all pairs of predictors should be examined. However, this will be very time consuming and, particularly for the case of association studies where LD decays with distance (Pritchard and Przeworski, 2001), a largely unnecessary process.

As well as improving performance, I personally feel that filtering provides an element of standardisation. For an experimenter, it must be easy to overlook possible biases introduced by the choice of predictors. Returning to the HapMap example, suppose we were particularly interested in searching for an association in the immediate vicinity of the `MTHFR` gene. This might lead us to consider additional SNPs in this region. Suppose then, a uniform prior probability of association was assigned, judging each SNP equally likely to be causal. By typing additional variants near to the gene, we would have implicitly increased the prior probability that an association lies in this region, which is dangerous if we then use the results to support a hunch that the causality was close by. Filtering the SNPs based on correlation should reduce this concern, as the density of those remaining will more closely match the profile of variation present.

3.5 Diagnosis of Results

As with any MCMC based method, *Sparse Partitioning*'s performance depends on the accuracy with which its Markov chain converges. The most straightforward and perhaps best means of diagnosis is to perform the analysis multiple times and compare each set of results. If the sets are in close concordance, this suggests reasonable convergence has been achieved. Of course, this check is by no means fail-safe. The method might each time end up in a local maximum, which is not necessarily a fair reflection of the posterior distribution as a whole. To guard against this, it is common to perform these runs from different starting states. Unfortunately, for the the case of *Sparse Partitioning*, such an approach is of little value and can only be misleading. Because of the sparsity involved, the majority of partitions will be considered very poor fits to the data. Therefore, in almost all cases, the method's first moves will be to revert back to the null model.

To judge whether a individual run has achieved stability, it is customary to examine trace plots. To this end, *Sparse Partitioning* keeps track of the posterior score and size of each partition visited. Raw plots of these sequences should highlight any obvious lack of convergence.

3.5.1 Cross-Validation

Cross-validation can be used both to compare the performance of different models and determine suitable parameter choices. To perform cross-validation, a method is first applied to a "training" set of samples, which it uses to predict response values for a "test" dataset. The method is then scored according to how well each predicted value $\hat{Y}_i$ compares with its actual value Y_i. If the response is continuous, a scoring system often used is mean squared residual error, $\sum_i (\hat{Y}_i - Y_i)^2/n$. If the response is binary, either a likelihood based measure can be devised or, if the predicted values are binary, the method can be scored according to the number of misclassifications, $\sum_i \mathbf{1}(\hat{Y}_i \neq Y_i)$.

The prediction score will depend on the choice of training and test samples so, unless a divide is obvious, it is common to repeat this process for a number of splits and average. If the training and test datasets are not chosen in a systematic manner, the final average will depend on how they were picked. Leave-one-out cross-validation (LOOCV) addresses this concern by considering only test datasets contain a single sample, then averaging over all n possible combinations.

I touched upon the way *Sparse Partitioning* handles cross-validation when explaining how the method copes with missing response values. Suppose Y_U corresponds to the response values in a test dataset. If the user sets each of these to missing, then *Sparse Partitioning* will estimate their values based on the posterior predictive distribution $\mathbb{P}(Y_U|\boldsymbol{X}, Y_O)$. Cross-validation can become a lengthy process, as the analysis of each training set will usually take almost as long as analysing the entire dataset. Therefore, by default, *Sparse Partitioning* performs "pseudo" LOOCV. How this differs to (true) LOOCV will become clear when I explain the calculations involved.

To perform LOOCV, the main algorithm remains unchanged. *Sparse Partitioning* continues to search for the posterior distribution of partitions using the entire dataset $\boldsymbol{X}$ and $\boldsymbol{Y}$. Additionally, at each iteration, the method calculates for each sample $\mathbb{E}(Y_i|\boldsymbol{X}, Y_{-i}, \mathbb{G})$, the expected value of the response given the current partition and the other response values. For a continuous response, this value is equal to $\mathbb{E}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G})$ when $U = \{i\}$ and O is its complement. Letting $\boldsymbol{\Sigma}$ again denote the inverse variance matrix of $\mathbb{P}(Y_U, Y_O|\boldsymbol{X}, \mathbb{G}, \sigma^2)$, this value equals $-(\Sigma_{UU})^{-1}\Sigma_{UO}Y_O = -\sum_{j\neq i}\Sigma_{ij}Y_j/\Sigma_{ii}$. At the end of the iterations, the set of values relating to sample i are averaged to provide a Monte Carlo estimate of the mean of the

posterior predictive distribution $\mathbb{P}(Y_i|\boldsymbol{X}, Y_{-i})$.

With a probit link function, near identical calculations return the value $\mathbb{E}(Z_i|\boldsymbol{X}, Z_{-i}, \mathbb{G})$ and in turn an estimate for $\mathbb{E}(Z_i|\boldsymbol{X}, Z_{-i})$. This value is converted into a probability that the corresponding response equals 1, which in turn can be dichotomised to determine classification error.

When a logit link function is used, $\mathbb{P}(Y_i = 1|\boldsymbol{X}, Y_{-i}, \mathbb{G})$ is calculated manually:

$$\mathbb{P}(Y_i = 1|\boldsymbol{X}, Y_{-1}, \mathbb{G}) = \left(1 + \frac{\mathbb{P}(Y_i = 0|\boldsymbol{X}, Y_{-i}, \mathbb{G})}{\mathbb{P}(Y_i = 1|\boldsymbol{X}, Y_{-i}, \mathbb{G})}\right)^{-1} = \left(1 + \frac{\mathbb{P}(Y_i = 0, Y_{-i}|\boldsymbol{X}, \mathbb{G})}{\mathbb{P}(Y_i = 1, Y_{-i}|\boldsymbol{X}, \mathbb{G})}\right)^{-1},$$

where the fraction is the ratio of the partition scores calculated with Y_i set to 0 and 1. Again, the posterior mean can be dichotomised to indicate the method's best guess for each response.

3.5.2 Permutation Tests

Permutation tests can be used to assign significance to *Sparse Partitioning*'s posterior probabilities. If the user reruns the analysis having first permuted the response values, the resulting posterior estimates will be obtained under a null hypothesis of no true association. By repeating this process, a p-value can be obtained for the largest marginal posterior probabilities, assessing the likelihood of obtaining values at least as large by chance alone. In particular, it is possible to obtain significance thresholds for the first, second and third strongest associations, and so on. As always, the resolution and accuracy of these p-values will be limited by the number of permutations. Although each permutation involves repeating the entire analysis, the time this takes will often be considerably shorter; when there are no true associations, the size of the current partition should remain small, greatly reducing the number and complexity of partitions which must be considered at each step of the sampling.

Gauging the significance of interaction probabilities is less straightforward. By permuting the response values, it is possible to obtain thresholds for each of the top pairwise probabilities. Unfortunately, these will often prove misleading. We are probably more interested in the conditional probability that these predictors interact, given their marginal probabilities of association. Consider a situation where *Sparse Partitioning* has found strong evidence that predictors 1 and 2 contribute and returns a probability of 0.45 that they interact. The importance of this finding depends on the states visited by the Markov chain. If, during the chain, $S_{\boldsymbol{I}}$ consistently equalled $\{1, 2\}$, then the prior probability these two predictors interact, given that they are the only two associations, is 0.5, so this becomes a weak result. By contrast, if $S_{\boldsymbol{I}}$ regularly contained a third predictor, the prior probability of their interaction would be 0.4, so this becomes a strong result.

In theory, we might consider conditional permutation (Anderson and Ter Braak, 2003; Werft and Benner, 2010), which attempts to generate datasets that preserve marginal effects while destroying true interactions. This would have to be carried out on a per-iteration basis, as this is the only time *Sparse Partitioning* provides an exact model for the underlying relationship. However, I don't believe this approach would be particularly informative, at least not compared to the extra processing it involves. Fortunately, I do not consider this issue a major concern. I view *Sparse Partitioning* as a method primarily aimed at detecting associations, whose performance is improved by taking nonlinearity into account. Therefore, the declaration of possible interactions is of secondary importance. Saying this, when describing the deterministic version of *Sparse Partitioning* in Chapter 5, I explain a strategy for obtaining alternative interaction probabilities by calculating Bayes factors conditional on the set of associations. I believe these Bayes factors examine interactions more closely.

I return to the issue of diagnosis in the next chapter, when applying *Sparse Partitioning* to real datasets.

3.6 Immediate Extensions

While I have described *Sparse Partitioning* for the case of a continuous or a binary response, the method is not limited to these two situations. The main three sampling stages involve $\boldsymbol{X}$ and $\boldsymbol{Y}$ only when scoring partitions. Each score is a composite of the partition prior, which is invariant of the regression model and precalculated, and the marginal likelihood, which depends on the choice of link function and the response assumptions. Therefore, to adapt the method for different response types, it is necessary only to calculate an alternative marginal likelihood.

As *Sparse Partitioning*'s regression model is similar in form to the generalized linear model, this proves useful when considering possible extensions. For example, when the response records count data, the natural assumption is a logarithmic link function and a Poisson likelihood assumption. If the new functions provide either a closed form for the marginal likelihood or a numerical approximation, this expression can be substituted in immediately. When such a form is not available, the marginal likelihood can instead be replaced with any measure of a partition's fit, as is the case when r is set to zero and the maximum likelihood is used in its place. However, as I discussed, I consider this reduced Bayesian alternative sub-optimal and advise against its use when possible.

One of my immediate objectives is to integrate *Sparse Partitioning* into the programming

language R (R Development Core Team, 2008), which would hopefully increase both the method's exposure and usability. It might be desirable to increase the level of user interaction, for example, to allow personalised measures of fit to be integrated within the code. While R is well-suited for this task, *Sparse Partitioning*'s computational requirements demand a high-level programming language, so I would probably have to investigate other formats. In the meantime, it is prudent that I develop the method for as many data types as possible.

3.6.1 Multiple Responses

Here, I consider the case when each sample is recorded for a number of responses. Such a set-up is becoming increasing common in genetics. Consider a fine-scale association study. Generally, the majority of the study's expense will be incurred obtaining the samples and typing the variants; relatively speaking, the cost of measuring samples for additional responses will be very slight. Therefore, it is efficient if a number of phenotypes can be investigated for a common set of predictors, a strategy adopted by Atwell *et al.* (2010). Studies of this type might become increasing popular once next-generation sequencing becomes commonplace — when "complete" typing is routine, there will be less need to choose an individual set of predictors for each response.

For this subsection only, I suppose each sample has been recorded for M response values. Let the vector $\boldsymbol{Y}_m = (Y_{1m}, Y_{2m}, \ldots, Y_{nm})^T$ contain the values for the mth response. The response values are now contained in a matrix: $\boldsymbol{Y} = [\boldsymbol{Y}_1\ \boldsymbol{Y}_2 \ldots \boldsymbol{Y}_M]$ (size $n \times M$). The aim is to investigate the partitions $\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M$, or, equivalently, the indicator vectors $\boldsymbol{I}_1, \boldsymbol{I}_2, \ldots, \boldsymbol{I}_M$, which correspond to the underlying relationships for each response.

The combined posterior distribution becomes $\mathbb{P}(\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M | \boldsymbol{X}, \boldsymbol{Y})$. However, the simplest approach is to regress each response on the predictors separately, at each stage considering only $\mathbb{P}(\mathbb{G}_m | \boldsymbol{X}, \boldsymbol{Y}_m)$. This is *Sparse Partitioning*'s default strategy, and will give the same results as constructing M separate experiments and running the method once for each.

If the sets of response values can be considered related, a natural question to ask is whether certain predictors influence more than one response. For example, we might imagine that a single biological pathway is able to affect the expression levels for a group of connected genes. This belief can be investigated *post-hoc* by analysing each response independently and looking for similarities between the M sets of posterior estimates. Alternatively, *Sparse Partitioning* provides the option to analyse the responses simultaneously, incorporating this idea as prior information.

In this multiple response set-up, I initially considered the collection of vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M$,

where $\boldsymbol{p}_m = (p_{1m}, p_{2m}, \ldots, p_{Nm})$ denotes the probabilities of association for the mth response. Analysing each response separately has the effect of supposing each p_{gm} is independent. If we believe it more likely that some responses are influenced by sets of common predictors, this implies that $p_{g1}, p_{g2}, \ldots, p_{gM}$ are in some way connected. This suggests a hierarchical prior of the form

$$\mathbb{P}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M) = \prod_g \left(\mathbb{P}(\zeta_g) \prod_m \mathbb{P}(p_{gm}|\zeta_g) \right),$$

where the variables ζ_g represent (vectors of) hyperparameters and are supplied with their own priors. The idea is that while $p_{g1}, p_{g2}, \ldots, p_{gM}$, the probabilities that different responses are influenced by predictor g, are free to vary, their prior distributions are linked by ζ_g, so providing the desired connection.

Similar to the univariate case, I argue that allowing p_{gm} complete flexibility is unnecessary, as it adds no extra value to the set-up. As before, I will shortly decide to fix these probabilities to their prior mean, which will have the effect of removing the second product term. For consistency, I will rename the variable ζ_g to p_g, though this will have a slightly different meaning compared to the univariate case.

Given the prior probabilities of association, the priors for each partition will be independent:

$$\begin{aligned}\mathbb{P}(\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M|\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M) &= \prod_m \mathbb{P}(\mathbb{G}_m|\boldsymbol{p}_m) \\ &= \prod_m B(\mathbb{G}_m)^{-1} \left(\prod_{g \in S_{\boldsymbol{I}_m}} p_{gm} \prod_{g \notin S_{\boldsymbol{I}_m}} (1 - p_{gm}) \right).\end{aligned}$$

Suppose complete flexibility was allowed. The conditional posterior distribution of p_{gm} would, like before, be determined solely by whether or not predictor g was declared associated for the mth response:

$$\begin{aligned}&\mathbb{P}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M|\boldsymbol{X}, \boldsymbol{Y}, \mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M) \\ &\qquad \propto \mathbb{P}(\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M|\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M) \times \mathbb{P}(\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M),\end{aligned}$$

and thus

$$\begin{aligned}&\mathbb{P}(\boldsymbol{p}_{gm}|\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{m-1}, \boldsymbol{p}_m^{-g}, \boldsymbol{p}_{m+1}, \ldots, \boldsymbol{p}_M, \boldsymbol{X}, \boldsymbol{Y}, \mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M) \\ &\qquad \propto p_{gm}^{\mathbf{1}(I_{gm} \neq 0)} (1 - p_{gm})^{\mathbf{1}(I_{gm} = 0)} \times \mathbb{P}(p_{gm}|\zeta_g),\end{aligned}$$

where $\boldsymbol{p}_m^{-g}$ represents $\boldsymbol{p}_m$ with element g removed and I_{gm}, the gth element of $\boldsymbol{I}_m$, indicates to which group of $\mathbb{G}_m$ predictor g belongs.

We could again consider what would happen if each p_{gm} was assigned a beta distribution prior, with shape parameters provided by the hyperparameter ζ_g. The posterior distribution of interest would become $\mathbb{P}(\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M, p_{11}, \ldots, p_{gm}, \ldots, p_{NM} | \boldsymbol{X}, \boldsymbol{Y})$. We would find that, just as in the univariate case, it would be possible to integrate across the probabilities, and so the Monte Carlo estimate of the posterior mean of p_{gm} would be a simple function of the Monte Carlo estimate of $\mathbb{P}(I_{gm} \neq 0)$, obtained by fixing each p_{gm} to its prior mean and sampling from $\mathbb{P}(\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M | \boldsymbol{X}, \boldsymbol{Y})$. Note that, as the prior distribution of p_{gm} is a function of ζ_g, this implies that p_{gm} is constant across responses, but will remain free to vary across predictors.

My conclusion contradicts the findings of Carvalho *et al.* (2008), the article which prompted me to consider hierarchical priors in the first place. However, I believe this discrepancy owes to the slight difference between my set-up and theirs. Essentially, their method, which is designed with far more responses in mind, has each associated predictor contributing to all responses. For this reason, a two-stage hierarchical prior is required, to reflect that the magnitude of this predictor's contribution varies greatly across responses. My method allows for a predictor to contribute only to a subset of responses, so this second level is not required. In effect, it is contained within the prior for the partitions.

As the set of probabilities corresponding to a particular predictor are now equal, I decided to set $p_{g1} = p_{g2} = \ldots = p_{gM} = p_g$. Compared to the univariate set-up, p_g takes on a slightly different meaning. It now represents a baseline probability that the gth predictor in some way influences the set of responses. Should, for example, its value increase during the MCMC sampling, this means that, for the next iteration, the prior probability that predictor g is associated with each response will be higher.

It remains to decide a prior for each p_g. I decided to opt for a beta distribution: $\mathbb{P}(p_g) = \beta(a_g, b_g)$. By setting $a_g = 1$, the value of b_g will be determined by the user's belief in each predictor's prior probability of having an impact. With the prior specified, the posterior distribution takes the form

$$\begin{aligned}\mathbb{P}(\mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M, &p_1, p_2, \ldots, p_N | \boldsymbol{X}, \boldsymbol{Y}) \\ &= \mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}, \mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M) \times \prod_m \mathbb{P}(\mathbb{G}_m | p_1, p_2, \ldots, p_N) \times \prod_g \mathbb{P}(p_g).\end{aligned}$$

At this point, I assume the sets of response values are independent, a decision I return to shortly. The overall marginal likelihood $\mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}, \mathbb{G}_1, \mathbb{G}_2, \ldots, \mathbb{G}_M)$ becomes the product of each individual response's likelihood $\mathbb{P}(\boldsymbol{Y}_m | \boldsymbol{X}, \mathbb{G}_m)$, making implementation straightforward. The

main sampling stages can be carried out concurrently, across multiple processors if desired. The method needs only to regroup once per iteration in order to resample p_g and, if necessary, missing predictor values. With each p_g assigned a beta prior, its conditional posterior distribution will also be beta, with shape parameters $1+\sum_m \mathbf{1}(I_{gm} \neq 0)$ and $b_g+M-\sum_m \mathbf{1}(I_{gm} \neq 0)$, where $\sum_m \mathbf{1}(I_{gm} \neq 0)$ counts the number of responses in which predictor g is currently declared associated.

For sparse problems, b_g is large and it is easy to understand the effect of this set-up. When a predictor is declared associated by one partition, the probability it is associated with a different one effectively doubles; if declared associated by two partitions, its probability trebles, and so on.

Simultaneously analysing responses is certainly the correct approach when one or more predictor values are missing. Were the responses considered separately, this would not appreciate that missing data should be consistent across experiments. For example, the first response might consistently sample state 0 for a particular unobserved predictor, while the second response might tend towards state 1. By simultaneously considering all responses, *Sparse Partitioning* resamples predictor values based on their combined conditional posterior distributions, taking into account all sets of responses.

So far, the restriction that $p_{g1}, p_{g2}, \ldots, p_{gM}$ are equal has not proved an issue, as I have not yet been in a situation where I would prefer this otherwise. However, it is conceivable that a user might consider, say, a predictor more likely to contribute to one response than to another, or for one response to have more associations than the rest. To accommodate such beliefs, I could let $p_{g1}, p_{g2}, \ldots, p_{gM}$ equal (fixed) functions of p_g, determined according to the user's belief in their relative individual values. Depending on the form of these functions, a revised proposal distribution for p_g might be required, but otherwise the remainder of the method would operate the same.

Correlated Responses

A current issue I face is what to do when the responses are highly correlated. Consider a situation where two sets of response values are very similar. Any partitions scoring highly for one will likely score highly for the other, boosting the posterior estimates of the predictors involved. However, as for the case of sample relatedness, this support is likely to be unfounded. It is more probable that the two response values are measuring very similar traits or alternative ways to measure the same trait.

At the moment, I wonder whether data reduction methods could be used to deal with this

problem (Fodor, 2002). For example, Ian Saunders suggested I applied principal component analysis to produce M linear combinations of the original response values. Each pair of these "new" responses will be orthogonal, allowing independence to be restored. Early testing indicates this approach might run into difficulties when nonlinearity is involved. If each response's underlying relationship is linear, then the signal might be carried over into principal components. However, if the relationship is nonlinear, any linear combination of responses seems likely to disrupt it.

Therefore, I feel the correct strategy might require explicit consideration of the correlations of responses. For the continuous case, it would be straightforward to set up a joint distribution from which each $(Y_{i1}, Y_{i2}, \ldots, Y_{iM})$ represents a sample, an inverse Wishart prior on the covariance matrix being an obvious choice. However, I foresee the calculations involved becoming unmanageable for even a few responses.

Perhaps a better strategy is to apply partitioning to the responses as well as the predictors. Zhang *et al.* (2010) implement an approach similar to that I have in mind. They group the responses into common modules which have similar patterns of values. In a similar vein, Stephens (2010) has been working on a method which divides the responses into not associated, directly associated and indirectly associated. The last category reflects that a predictor-response pair might appear associated, but this is a consequence of the response being highly correlated with another.

3.6.2 Parallelisation

Speed is of crucial importance in MCMC methods. The faster each iteration, the greater the number that can be performed, so the more reliable the results. In *Sparse Partitioning*, computation of the marginal likelihood accounts for over 95% of all processing time, so I have put much thought into trying to optimise this process. As mentioned, it proves convenient that each group's design matrix $\boldsymbol{J}_k$ is sparse with only a single 1 in each row, as this speeds up calculation of the matrix $\boldsymbol{B}$. However, the overwhelming bottle-neck in the algorithm is the solving of $\boldsymbol{BA} = \boldsymbol{J}^T\boldsymbol{Y}$, made worse when confounding cofactors are introduced. For most changes proposed to the current partition, only a single non-null group is affected, so this suggests a trick similar to inversion by partitioning (Press *et al.*, 2007) might be incorporated to speed up the operation.

It is convenient that Sampling Stage Two can immediately be parallelised. This stage requires exhaustive calculation of the neighbourhood of all partitions obtainable by a change to one component of $\mathbb{G}$. This search can simply be assigned across processors. Typical of most MCMC algorithms, Sampling Stage One is not easily adapted for parallelisation, as the

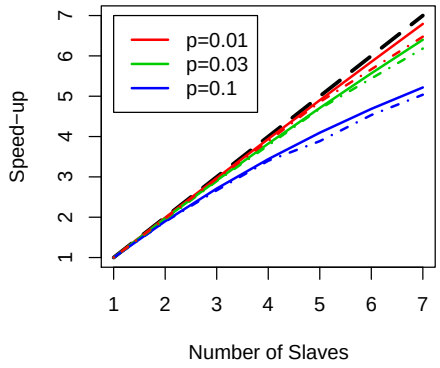


Figure 3.2: *Parallel processing of MCMC samplings. During the resampling of* $\boldsymbol{I}$*, the master processor simultaneously instructs each slave to resample group membership for a different predictor. The main text explains when each slave's resampling will be valid. The expected number of valid samplings after looping once through the slaves depends on* p*, the likelihood that a changed value of* I_g *is reported. The solid coloured lines represent the theoretical speed-ups; the dashed coloured lines some speed-ups observed in practice. For comparison, the black dashed line represents a perfect linear speed-up.*

sampling of step $t+1$ relies on knowing the outcome of step t.

However, for high-dimensional problems, I devised a system which takes advantage of the likely sparse nature of the problem (Speed, 2008). Suppose we have H slave processors available and assume the order in which predictors are examined has not been shuffled. The master processor instructs the $(h{+}1)$th slave to sample I_{g+h+1} using the current values of $I_{g+1}, I_{g+2}, \ldots, I_{g+h}$. The sampled value of I_{g+h+1} will be valid only if the first h slaves make no changes to the current model. If one of these slaves makes a change, it will be necessary to backtrack and ignore all samplings after the point at which this occurred.

Fortunately, when the number of associations is small compared to N, there is a high probability that each step will make no change to the current partition. In particular, if most predictors remain not associated, this parallelisation is very efficient. Suppose the average probability that a step changes the current partition is p. By considering a (truncated) geometric distribution with probability of success p, it is straightforward to calculate a theoretical bound on the improvement possible. In Figure 3.2, each solid coloured line represents the maximum speed-up factor achievable for a particular value of p. By comparison, the dotted coloured lines represent speed-ups I achieved in practice using an eight-core processor. We can see that even for moderate values of p, corresponding to probabilities of association of between 1 in 10 and 1 in 100 (high by sparse standards), the increase remains close to linear. A similar implementation, called "Speculative MCMC", has been proposed by Byrd *et al.* (2008, 2010).

Chapter 4

Testing and Applications

This chapter is divided in two. The first section uses simulated datasets to compare the performance of Sparse Partitioning *with existing methods. This kind of testing, where the underlying relationship can be specified in advance, proved invaluable during development of the methodology. The second section applies the method to three association study datasets.*

4.1 Simulation Studies

In Chapter 2, I presented brief results from Study One. This study acted as the template for the simulation studies. In total, I performed nine further studies, each of which altered a certain aspect of the set-up:

Study One	100 samples, 1000 binary uncorrelated predictors, continuous response.
Study Two	Causal predictors unobserved.
Study Three	10% of predictor values missing.
Study Four	Non-normal noise.
Study Five	Tertiary predictors, 2 causal loci.
Study Six	Binary response, 3 or 4 causal loci.
Study Seven	Correlated predictors, 4 causal loci.
Study Eight	Examine effect of prior choice.
Study Nine	Examine effect of number of iterations.
Study Ten	Non-disjoint underlying relationship, 3 or 4 causal loci.

To offer the fairest assessment of *Sparse Partitioning*'s performance, I tried to test it in as many scenarios and against as wide a range of existing methods as possible. However, I omitted some methods from comparison. Some are for reasons of duplication. As I discussed in the introduction, there are a wide range of methods in the multiple regression category ($K > 1;\ S = 1$). While, fundamentally, these propose the same underlying relationship,

frequentist approaches vary in their choices of penalty functions, while Bayesian approaches differ in their priors. I picked Shotgun Stochastic Search (*SSS*) to represent this category of methods. Being Bayesian, it seems more relevant and, unlike the others methods I mentioned, its finalised code is readily available. Furthermore, *Sparse Partitioning* has many similarities to *SSS* when the maximum group size is set to 1, which made for interesting comparison.

Generally, a regression method dealing with a continuous response can be used to analyse a binary response, either directly or through introduction of a suitable link function. The converse is not true. I consider *Sparse Partitioning* best-suited for experiments involving a continuous response. Almost always, a continuous response value will provide more information than a binary one. Nonlinear regression methods rely far more on information in the data than linear methods. This is a consequence of their relative complexity; the degrees of freedom of an interaction model has the potential to grow exponentially with the number of predictors involved. For these reasons, I excluded from comparison the methods set up to only handle binary response values.

In order to compare methods, I asked each to declare a fixed number of associations, then counted how many of the causal predictors it correctly found. For example, for studies in which there were three causal predictors, I required each method to return the three predictors for which it found most evidence of an association, then scored according to how many of these three were simulated to be causal. From these figures, I computed an "average detection accuracy" for each scenario by averaging the score over 100 datasets.

It would have perhaps been more obvious to compare methods according to detection power. In this case, I could have placed no limit on how many associations a method could declare, then simply scored each by the number of causal predictors it discovered. However, I foresaw a number of difficulties with such an approach. In particular, it would have been necessary to decide a threshold for each method, beyond which predictors were declared associated. This had the potential of producing misleading power estimates; for example, if I had set a method's threshold too conservatively, it would have been at a disadvantage. As an alternative, I could have used cross-validation to determine thresholds, but this would have had to been done on a per-dataset basis, so would have been incredibly time consuming.

4.1.1 Generating Datasets

Each study considered a range of scenarios, each of which examined a different combination of underlying relationship and "causal predictor frequency". When a causal predictor is binary, this frequency denotes how often it takes value 1 in the general population. If the predictors are thought of as SNPs, this value corresponds to the minor allele frequency.

Using Study One as an example, I explain the basic methodology I used to produce one dataset for a particular scenario. This study considered three different underlying relationships, the first of which was $f(\boldsymbol{X}) = f(X_1, X_2, X_3) = X_1 + 1.5X_2 - 2X_3$. To generate other datasets, I changed $f(\boldsymbol{X})$ accordingly. Let $\boldsymbol{F}$ denote the set of unique possible values of $f(X_1, X_2, X_3)$, which for binary predictors contains the images of members of $\{0, 1\}^3$.

I generated datasets in a retrospective manner, first simulating a sample's response, then its predictor values. The first step was to sample F at random from $\boldsymbol{F}$, from which I obtained y, a response value realisation, through addition of normally distributed noise: $y = F + \mathbb{N}(0, \sigma^2)$. This process ensured the response values were sampled evenly across their full range, rather than according to their prevalence. From this response, I generated x_1, x_2, x_3, values for the three causal predictors, according to

$$\begin{aligned}\mathbb{P}(x_1, x_2, x_3|y) &\propto \mathbb{P}(x_1, x_2, x_3) \times \mathbb{P}(y|x_1, x_2, x_3) \\ &\propto m_1^{x_1} m_2^{x_2} m_3^{x_3} \times \exp\left\{-\tfrac{1}{2\sigma^2}(y - f(x_1, x_2, x_3))^2\right\},\end{aligned}$$

where m_1, m_2 and m_3 are the causal predictor frequencies relating to X_1, X_2 and X_3. I considered five possible values for m_g: 0.05, 0.1, 0.2, 0.4 or "random". For the first four values, m_1, m_2 and m_3 were set equal; for the fifth, each m_j was sampled uniformly at random from the interval $[0.05, 0.95]$. For each model, I chose σ^2 so the proportion of observed variation explained by the true underlying relationship ranged from about 0.1 to 0.5.

For each of the 997 remaining predictors, I sampled its values independently from a Bernoulli distribution, such that $\mathbb{P}(X_{ig} = 1) = \eta_g$, where η_g was drawn uniformly at random from the interval $[0.05, 0.95]$. The last step was to reorder the predictor set, just in case a method benefited by the causal predictors being located at the start.

For an association study, the proportion of variance explained corresponds to the heritability, so my range of values might seem unrealistically high. To some extent, such a choice was necessary. I found 100 samples and 1000 predictors to be a suitable size for testing "large p, small n" datasets in a reasonable time. Consider from a frequentist point of view, the amount of variation which much be explainable to achieve significance for different degrees of freedom. To obtain a p-value less than 10^{-3} for a sample of size 100, a model with degrees of freedom 1, 2, 3 or 4 must explain at least 10, 13, 15 or 17% of the variation, respectively (I explain how I arrive at these figures very shortly). For a significance threshold of 10^{-4} these values rise to 14, 17, 19 and 21%. This demonstrates the heritability required when the sample size is moderate. By comparison, were 1000 samples available, a 5 degrees of freedom model could achieve genome wide significance (p-value $< 10^{-6}$) by explaining less than 3.5% of the

variation. Reassuringly, I have come across countless datasets where variants explain upwards of 20% of variation, even across hundreds of samples. I provide some examples of these a bit later on.

4.1.2 Details and Settings for Methods

Most methods require the user to make some parameter choices, so as far as possible I tried to select these consistently. When using third-party software, any parameters I do not mention specifically were left at their default values. I ran four of the existing methods using their implementation in the programming language R (R Development Core Team, 2008).

Single

I designed *Single* to represent the most basic, yet popular, approach for analysing datasets. It is a one-predictor-at-a-time method, testing each individually for evidence of an association with the response. For predictor g, it considers the null model $f(X_g) = \theta_0$ compared to the alternative model $f(X_g) = \theta_{X_g}$. It makes the standard assumption that the residuals are normally distributed, with mean 0 and unknown variance σ^2, allowing it to calculate the maximum likelihood estimates under each hypothesis.

Single's test statistic is the ratio of these likelihoods. For the case of a continuous response, this test statistic can conveniently be expressed as $n \log(TSS_0/TSS_1)$, where TSS_0 and TSS_1 are the residual sums of squares under each model, calculated using the least squares estimates. To obtain a p-value for each predictor, this test statistic is compared to a χ^2 distribution with degrees of freedom 1 (binary predictors) or 2 (tertiary predictors). Notice that $1 - TSS_1/TSS_0$ represents the proportion of variation explained by the alternative model, which is how I arrived at the earlier estimates for required heritability.

Single also offers a Bayesian version, invoking the same prior choices for regression coefficients and σ^2 as *Sparse Partitioning*. Let $\mathbb{G}^0$ denote the null partition which declares no predictors associated and $\mathbb{G}^g$ denote the partition corresponding to $S_{\boldsymbol{I}} = \{g\}$. The marginal likelihoods under these null and alternative models will equal $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}^0)$ and $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}^g)$, which when the response is continuous will both take the form

$$r^{\frac{D}{2}}(2\pi)^{-\frac{n}{2}}|\boldsymbol{B}|^{-\frac{1}{2}} \times \Gamma(\tfrac{n}{2})(\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^T\boldsymbol{Y})^{-\frac{n}{2}},$$

with $\boldsymbol{B}$, $\boldsymbol{J}$ and D specific to each model. If we were to standardise the columns of $\boldsymbol{J}$ so that $\boldsymbol{J}^T\boldsymbol{J} = \boldsymbol{I}_D$, we would find that $\boldsymbol{B} = \boldsymbol{J}^T\boldsymbol{J} + r\boldsymbol{I}_D = (1 + r) \times \boldsymbol{I}_D$. Additionally, the posterior mode of $\boldsymbol{\Theta}$ could be shown to equal $(1 + r)^{-1}\boldsymbol{J}^T\boldsymbol{Y}$, a value obtained by scaling the least square estimate towards zero by a factor $(1 + r)$. This similarity continues when we consider

the logarithm of the Bayes factor comparing the two hypotheses:

$$\text{BF} = \frac{D' - D}{2} \log\left(\frac{r}{1+r}\right) \times \frac{n}{2} \log\left(\frac{\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{Y}/(1+r)}{\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{J}'\boldsymbol{J}'^T\boldsymbol{Y}/(1+r)}\right),$$

where $\boldsymbol{J}$ and $\boldsymbol{J}'$ are the design matrices corresponding to the null and alternative hypotheses. Bearing in mind that the residual sum of squares in the frequentist version takes the form $\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{Y}$, this highlights how similar the frequentist and Bayesian approaches are. For fixed r, the Bayes factor values for binary ($D' = 2$) or tertiary ($D' = 3$) predictors will be ordered the same as the frequentist test statistics, and therefore the p-values. When $\boldsymbol{J}$ is no longer standardised, this equivalence can be retained by shrewd prior choice, one which assigns higher prior variance to coefficients corresponding to less variable predictors (Wakefield, 2009). In the case of association studies, this assumes that rarer variants are likely to have larger effect sizes.

Furthermore, posterior odds are calculated by multiplying the Bayes factor by the prior odds $p_g/(1 - p_g)$, which in turn provides the posterior probability of association. Therefore, when each predictor is considered equally likely to be associated, the frequentist and Bayesian versions will essentially give the same results. As this is the case in the simulation studies, I only present results from the former. The main advantage of the Bayesian version comes when certain predictors are judged more or less likely to be associated. The method will then be able to incorporate this knowledge, potentially changing the ordering of posterior probabilities as a result.

When confounding variables are supplied through $\boldsymbol{\Omega}$, these are added into both the null and alternative hypotheses, which then take the forms $f(X_g) = \theta_0 + \boldsymbol{\Omega}\boldsymbol{\omega}$ and $f(X_g) = \theta_{X_g} + \boldsymbol{\Omega}\boldsymbol{\omega}$, respectively. For the frequentist version, the maximum likelihood test also calculates an estimate of $\boldsymbol{\omega}$ for each hypothesis; while for the Bayesian version, the marginal likelihoods for the null and alternative partitions are calculated with $\boldsymbol{\Omega}$ included.

At the moment, both forms of *Single* use independent tests. A Bayesian version has the capacity to compare more than two models at once, calculating a posterior probability for each. Assuming at most one predictor is associated, there are $1 + N$ possible models, so we could prescribe (relative) prior probabilities for each and assess which is most likely in light of the data. However, perhaps this would not be appropriate. Our use of one-predictor-at-a-time methods is mainly for convenience, not necessarily because we believe only one predictor is causal.

Pairs

Pairs extends *Single* to test all possible pairwise associations. The methodology is the same, except now the alternative hypothesis takes the form $f(X_g, X_{g'}) = \theta_{X_g X_{g'}}$. For binary predictors, this model will have 3 or 4 degrees of freedom. Again, the test statistic is a ratio of residual sums of squares, comparing how much the model fit improves by adding in two predictors. In a similar fashion, a p-value can be estimated by comparing this statistic to a χ^2 distribution with either 2 or 3 degrees of freedom.

Classification and Regression Trees

CART was run using its implementation for R, which is contained within the package `tree`. The function *tree* returned a model with an unrestricted number of associations, which I then reduced to the required size using *prune*.

Random Forests

RF was also run using its R implementation, contained in the package `randomForest`. The function *randomForest* returned importance weightings for each predictor, on which an ordering could be based.

Shotgun Stochastic Search

SSS was run using software developed by Hans *et al.* and hosted on the website of the Ohio State University's Department of Statistics. The prior belief in the number of associations, *priormeanp*, was set to 5, while the number of iterations, *iters*, was set to 100. The parameter *nbest* determines the number of top scoring models collected, conditional on which the posterior estimates are made. I set this to 1000. Generally, I observed that the top posterior probabilities dropped off very sharply, so I had no concerns regarding this value's specification.

Logic Regression

Logic was run using the R package `LogicReg`. The function *logreg*, with parameters $select = 2$ and $nleaves = s$, returned a logic tree of size s. The package offers the ability to run using an MCMC based method by setting $select = 7$. However, the size of the simulated datasets proved too large for R to handle, so I was unable to use this option. Relatively early on in my research, I coded my own MCMC version of *Logic*, correcting what I perceived to be errors in its set-up. At the time, I concluded such a version still had a number of short-comings, leading me to supercede that attempt with *Sparse Partitioning*.

Multivariate Adaptive Regression Splines

MARS was run using the R package `mda`. The function *mars*, with parameter choices $nk = 2s + 1$ and $degree = 3$, returned the best model with at most s predictors and allowing for at most three-way interactions.

Sparse Partitioning

Sparse Partitioning was run for 200 iterations, which typically took less than one minute. K, S and C were left at their default values (4, 4 and 3), while p_g was set to $5/N$. r was set to 1, which matched the default value of the corresponding parameter in *SSS*.

Graphical Representation of Results

With only two exceptions, all figures take the form of that presented in Chapter 2. Each has three columns, which correspond to the three underlying relationships being tested. Within each plot, the x-axis marks the causal predictor frequencies; the fifth of which ('?') indicates that each causal predictor's frequency was drawn uniformly at random from the interval $[0.05, 0.95]$. The y-axis reports the average number of causal predictors detected, except for Study Ten where the average proportion is used instead. Finally, each coloured line corresponds to a different method, generally one of the eight listed at the start of this chapter, and reports that method's detection scores for each different scenario.

4.1.3 Study One: Additional Results

As a reminder, the underlying relationships used here, and in many of the studies, were

Model	Underlying Relationship
I	$Y = X_1 + 1.5X_2 - 2X_3$
II	$Y = 1.5X_1 \times X_2 + X_3$
III	$Y = f(X_1, X_2) + X_3,$ where $f(0,0) = 0,\ f(1,0) = 1,\ f(0,1) = 2,\ f(1,1) = -1$

The top plot of Figure 4.1 is identical to that in Chapter 2, except it is just possible to make out the addition of a dotted black line. This corresponds to running *Sparse Partitioning* allowing only one copy of each predictor ($C = 1$). That the choice of C has very little effect on the results corresponds to a very pleasing discovery. Initially, I was hesitant to allow predictors to feature in multiple groups. Although I appreciated that such a situation might arise, I feared that allowing multiple copies would have a detrimental effect on most analyses when such an allowance was unnecessary. Once again, this finding backs up my view that

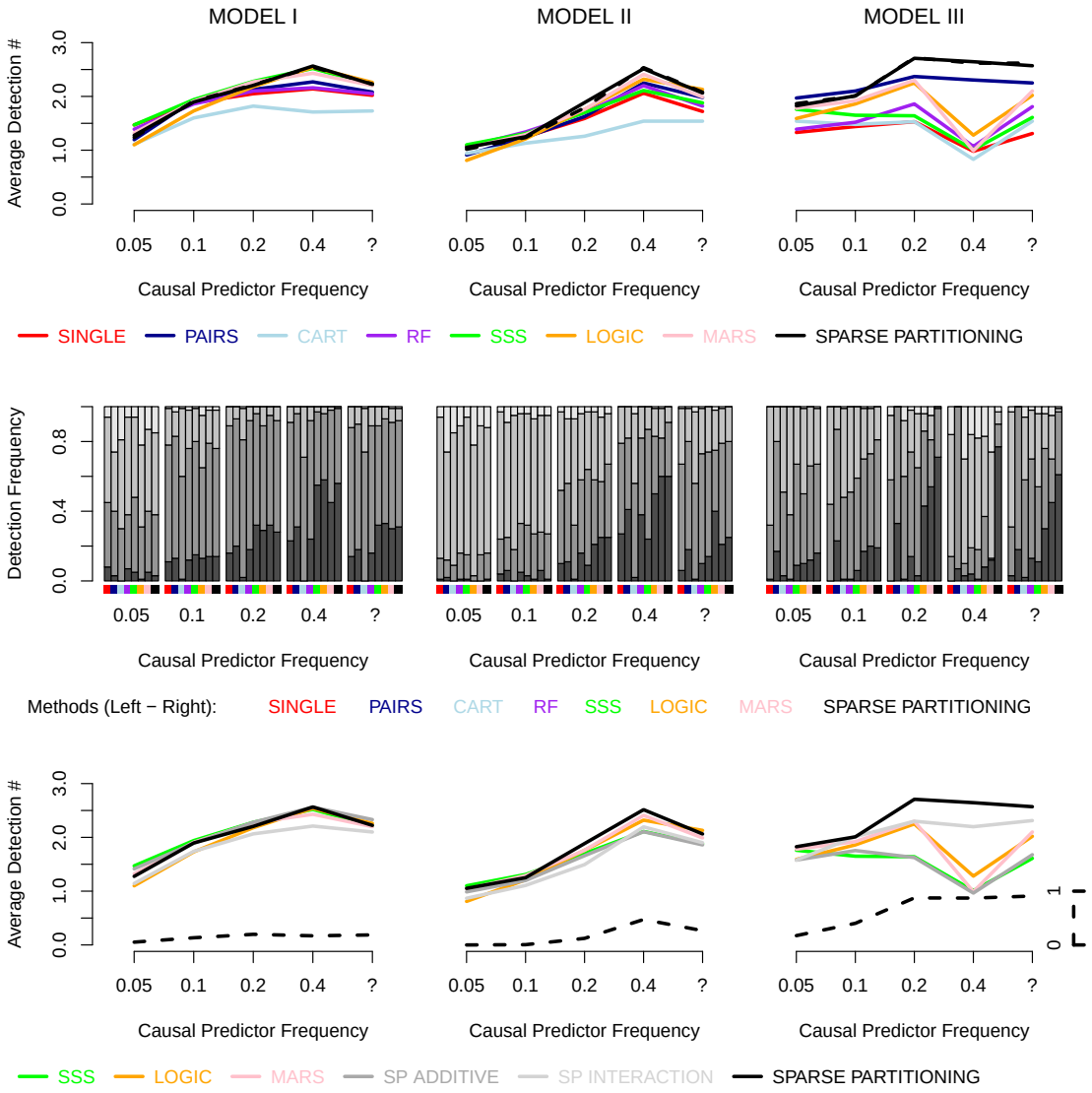


Figure 4.1: *Full results of Study One. The top row provides the average detection score for each method. These three plots are identical to those in an earlier chapter, except for the addition of a black dashed line, which presents the results of* Sparse Partitioning *when C, the maximum number of copies of each predictor, was set to 1. In the middle row, vertical bars provide the proportion of times each method correctly found 3, 2, 1 and 0 causal predictors (more details in main text). Similar to those in the top row, the three plots in the bottom row compare the performance of two new methods:* SP Additive *is* Sparse Partitioning *restricted only to additive models ($S = 1$); while* SP Interaction *is* Sparse Partitioning *restricted only to interactive models ($K = 1$). Additionally, the dashed lines mark the top pairwise interaction probabilities found by* Sparse Partitioning*, when run with its standard settings ($K = 4, S = 4, C = 3$).*

increased generality does not necessarily come at the expense of performance. If anything, the case when $C = 3$ seems to perform better. This can perhaps be explained by considering its impact on the MCMC search. When a predictor is not associated, it has no effect. But when the predictor is associated, it adds more flexibility into the MCMC sampling, providing a wider choice of moves that can be considered.

CART has performed poorly here, which is symptomatic of its performance in general. It is hard to diagnose why this is, as there does not appear to one type of dataset for which it performs especially badly. *CART* is designed for continuous predictor values, where thresholding has more relevance. This leads me to believe its method is optimised for this set-up and therefore suffers when used for categorical values.

The middle plot presents the same information in a different format. Each vertical bar refers to a particular method, for a specific underlying relationship and causal predictor frequency. It is composed of four segments which, from bottom to top, have lengths equal to the proportion of time the method correctly declared exactly 3, 2, 1 and 0 causal predictors. For example, we see that the frequency that all three causal predictors were detected (the black bars) generally increased as the causal predictor frequency rose. The heights of the bars mirror the average detection score, showing how *Sparse Partitioning* is best equipped as the model becomes more complex.

The bottom plot shows the effect of changing some of *Sparse Partitioning*'s input parameters. When $S = 1$ (*SP Additive*), the maximum group size is limited to one, so the method considers only additive models. When $K = 1$ (*SP Interaction*), only one tree is permitted, forcing the full interaction model to be fitted at each step. As expected, the performance of *SP Additive* was nearly identical to that of *SSS*, and their lines almost exactly coincide for Model III. Also to be expected, the performance of *Sparse Partitioning* was damaged when K is set to 1, as then the method necessarily overfitted the true model. However, it should be noted that *SP Interaction* was the second best performing method for Model III. This supports my belief that using an underlying relationship too general is less of a penalty than using one too restrictive.

Additionally, the dashed lines in the bottom plot mark the average of the highest posterior probability of interaction for the standard version of *Sparse Partitioning*. This provides an insight into *Sparse Partitioning*'s mechanics. For Model I, this line is very flat and close to zero, as desired when the true underlying relationship contains no interactions. For Model II, the line mirrors the detection accuracy; the point at which *Sparse Partitioning* began to detect the interaction is the point that it began to successfully detect all three predictors. The same effect is seen for Model III, except now the signal from the interaction was stronger, so

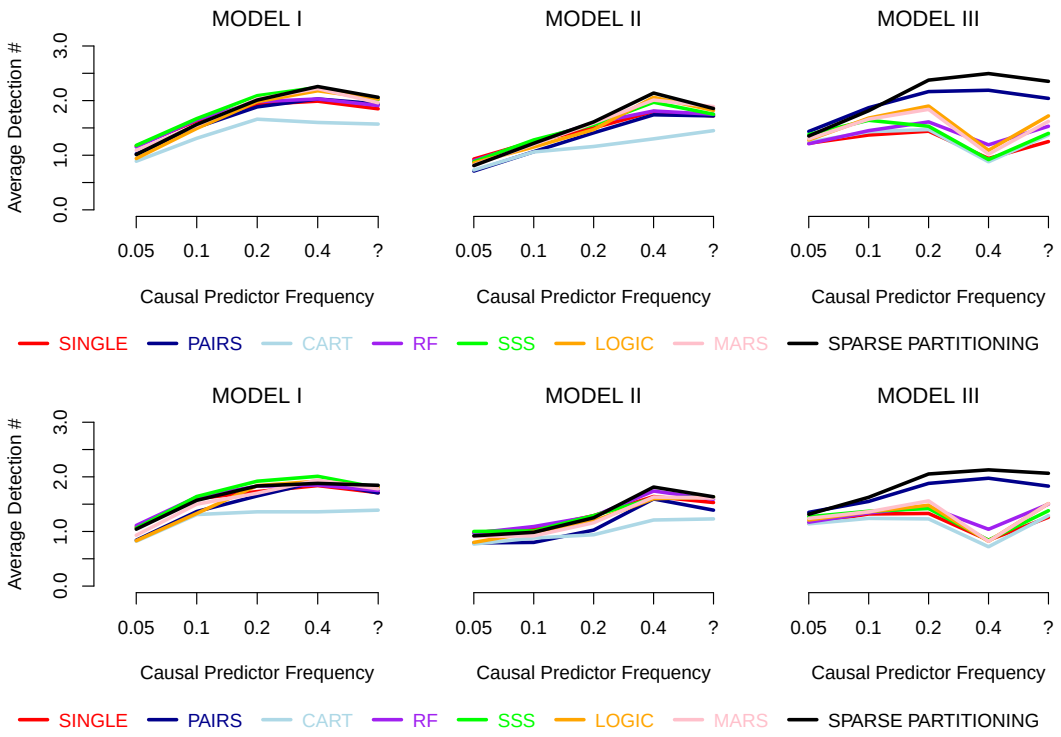


Figure 4.2: *Results of Study Two. The three causal predictors were unobserved, but each in high correlation (measured in terms of r^2) with an observed predictor. For the top plots $r^2 = 0.9$, for the bottom plots $r^2 = 0.8$.*

it was detected sooner.

I have not been able to understand why the majority of methods experienced a sharp drop in performance in Model III at the highest causal predictor frequency (0.4). It was suggested that this might be due to my retrospective choice of sampling. To test this theory, I tried creating comparable datasets prospectively, but the trend still appeared to persist (results not shown).

4.1.4 Study Two: Causal Predictors Unobserved

Due to high correlations between genetic variants, an association study is often able to infer the location of causal predictors even if they have not been typed directly. For this reason it is permissible, and usually more efficient, to type just a subset of variants. In this study, I considered the case when the causal predictors were not observed, but instead highly correlated with observed predictors.

To begin with, I generated data in the same manner as the previous study, obtaining for each sample a realisation for Y, then for X_1, X_2 and X_3. However, rather than using the values for X_1, X_2 and X_3 directly, I considered the introduction of three new predictors, X_1',

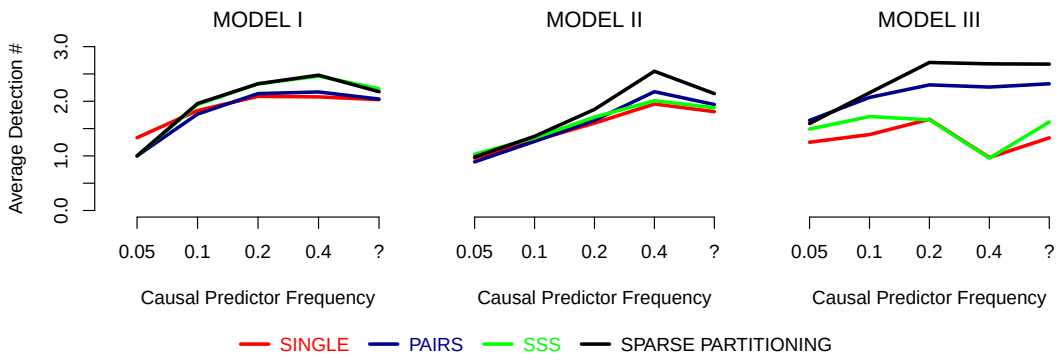


Figure 4.3: *Results of Study Three. 10% of predictor values were set to missing.*

X_2' and X_3', each of which tagged one of the causal variants. To generate, for example, values for X_1', I started with those generated for X_1, then randomly toggled values until the squared correlation between X_1' and X_1 dropped below the desired level (either r^2 equal to 0.9 or 0.8).

The results for the two levels of correlation are shown in Figure 4.2. The shapes of the plots closely match those of Study One, albeit, as expected, with lower average detection accuracy. Although the gap closes slightly, it is pleasing that *Sparse Partitioning* continued to outperform *Pairs*. When there are just two causal predictors, *Pairs* could be considered the gold standard method, as it performs an exhaustive search of all two-predictor models. Reducing the correlation between X_g and X_g', just like increasing σ_ε^2 has the effect of increasing the noise component of the model. Once the noise increases to the extent that one causal predictor becomes "unfindable", *Pairs* would be expected to perform at least as well as any other method.

4.1.5 Study Three: 10% of Predictor Values Missing

In this study, I tested how *Sparse Partitioning* would perform when confronted with incomplete data. Having generated datasets according to Study One, I then removed 10% of the predictor values at random. I omitted *CART*, *RF*, *Logic* and *MARS* from this study, as their implementations are unable to accept missing values. The results, shown in Figure 4.3, closely mirror the corresponding plots for Study One, indicating that the four remaining methods are fairly robust to moderate levels of missingness.

4.1.6 Study Four: Non-normal Noise

For a continuous response, *Sparse Partitioning* calculates a likelihood under the assumption of normally distributed residuals. Therefore, I tested the impact when this assumption had been violated. In this study, I simulated datasets using first exponential, then uniform noise. The

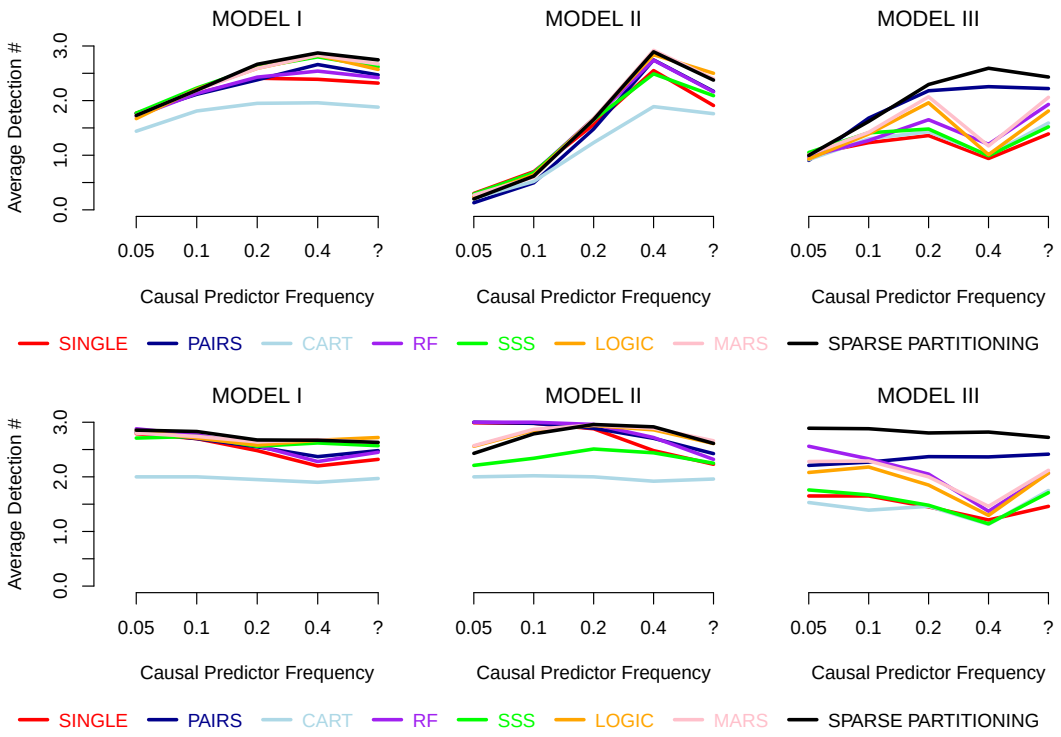


Figure 4.4: *Results of Study Four. For the top three plots, datasets were generated such that the residuals were distributed with exponential noise; for the bottom three plots the residuals were distributed uniformly.*

process used to generate each dataset was identical to that for Study One, except I changed $\mathbb{P}(y|x_1, x_2, x_3)$, the distribution of the residual values $y - f(x_1, x_2, x_3)$. For the two alternatives, I chose the distribution parameters (the rate of the exponential or width of the uniform) to produce heritabilities similar to those in Study One.

Figure 4.4 displays the results. The introduction of exponential noise, shown in the top row of plots, does not have a marked effect on the results; the plots still closely resemble those of Study One. This is not the case with uniform noise, where each model, and in fact each method, has responded differently to its introduction. Nonetheless, with the exception of the low frequency end of Model II datasets, *Sparse Partitioning* has maintained its lead, and has actually dramatically improved under Model III. These results must, to a large extent, owe to the nonlinear nature and flexibility of *Sparse Partitioning*. Because there are no restrictions on the shape of the underlying relationship, it is able to adapt $f(\boldsymbol{X})$ to better fit the assumption of normally distributed noise.

4.1.7 Study Five: Tertiary Predictors

For each sample, I created two causal tertiary predictors by first generating two pairs of binary values according to the method used in Study One, then summing these pairs. If each tertiary

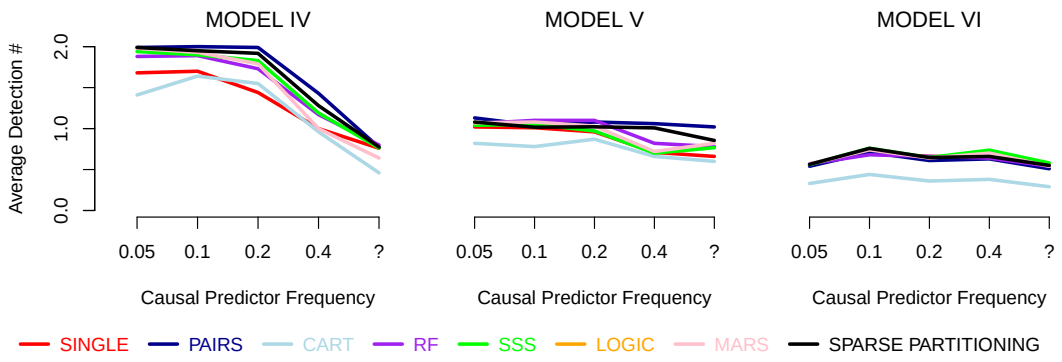


Figure 4.5: *Results of Study Five. Each predictor was tertiary, 2 of which were causal.*

predictor is viewed as an allele count across a homologous pair of loci, then, except for option '?', the causal predictor frequency will represent the minor allele frequency. Furthermore, because of the process used to generate predictors, the underlying state frequencies will be in Hardy-Weinberg equilibrium. (When the causal predictor frequency is '?', the minor allele frequency will be the average of the two randomly drawn frequencies, and Hardy-Weinberg equilibrium is unlikely to hold.)

I constructed the following three underlying relationships:

Model	Underlying Relationship
IV	$Y = I_{X_1>0} + I_{X_2>1}$
V	$Y = f_1(X_1) + f_2(X_2)$ with each $f_k(0)$, $f_k(1)$ and $f_k(2)$ chosen at random
VI	$Y = f_1(X_1) + f_2(X_2)$ with each f_k additive $(f_k(0) + f_k(2) = 2f_k(1))$

The three models considered were all additive between the two causal predictors. Model VI was also additive within the causal predictors, while Models IV and V were not. The results for each model are shown in Figure 4.5. Once again, *Sparse Partitioning* performed well, however, for random causal predictor frequencies it was overtaken by *Pairs*. The latter method is perhaps well-suited to this model, as it focuses on underlying relationships containing one pair of causal predictors. Considering that *Sparse Partitioning* allows for up to 16 causal predictors, it is very promising that it is only slightly outperformed. The shape of the plot for Model IV is peculiar, the only one which favours detection at lower causal predictor frequencies. This is a consequence of the very biased sampling of response values. There are only three distinct values for $f(X_1, X_2)$, so an equal chance of each response being centred on each. When the causal predictor frequencies are small, the response $Y = 1$ is very likely to relate to the configuration $(X_1, X_2) = (1, 0)$, while $Y = 2$ will probably correspond to $(X_1, X_2) = (1, 2)$, which allows easy distinction of the causal predictors. As the frequency increases, this distinction fades.

4.1.8 Study Six: Binary Response

A binary response generally contains less information than its continuous counterpart. Therefore, to maintain reasonable power for non-trivial models, I reduced the number of predictors and increased the number of samples. I mirrored the study of Mukherjee *et al.* (2009), both in the choice of underlying relationships and by generating datasets with 100 predictors and 200 samples. The only difference is that I also conditioned on individual causal predictor frequencies, while their study, in effect, set the frequency to '?' throughout. For each underlying relationship, Mukherjee *et al.* used a Boolean function to determine $\mathbb{P}(Y_i = 1)$; if the function evaluated true, Y_i was sampled using $\mathbb{P}(Y_i = 1) = 0.9$; if false, using $\mathbb{P}(Y_i = 1) = 0.1$. This corresponds to setting $f(\boldsymbol{X})$ to 2.2 or -2.2 when using a logit link function.

Model	Underlying Relationship
VII	$\mathbb{E}(Y) = 0.1 + 0.8 \times X_1 \wedge (X_2 = X_3)$
VIII	$\mathbb{E}(Y) = 0.1 + 0.8 \times (X_1 \wedge X_2^C) \oplus X_3$
IX	$\mathbb{E}(Y) = 0.1 + 0.8 \times (X_1 \wedge X_2) \oplus (X_3 \wedge X_4^C)$

($\wedge$ = AND and $\oplus$ = EXCLUSIVE OR.)

The results for each model are shown in Figure 4.10. I have kept the third graph on the same y-axis, as no method declared on average more than 3 causal predictors correctly. *Sparse Partitioning* and *MARS* were the best performing methods in this study, sharing the top two places across the three models. For *MARS*, its success in Model VIII is somewhat peculiar, as the method treats binary response values as if they were continuous. Perhaps this demonstrates that there is merit in analysing a binary response as if it was quantitative. Considering this approach will generally be faster, it can often serve as a useful first pass.

Sparse Partitioning's results were obtained through use of a logit link function. The reason that I implemented the probit link option, was the hope that it might speed up performance at limited expense. The black dashed lines in each plot indicate the results of *Sparse Partitioning* when this option was taken. The amount of speed-up offered by the probit choice will depend heavily on the nature of the partitions being scored. Certainly it is at least three times as fast, but even with relatively simple models, for example, partitions of size 3 or 4, the speed-up can be over ten-fold. When testing this version, I therefore afforded the method a few more iterations, selecting 600 as opposed to 200. This still represented a considerable reduction in processing time. As the plots show, the version with a probit link function performed favourably compared to the logit alternative, and therefore is my method of choice when the response is binary. Unfortunately, as I will point out in the next chapter, *Sparse Partitioning* is unable to use a probit link function exclusively, as there are occasions when this simply isn't possible.

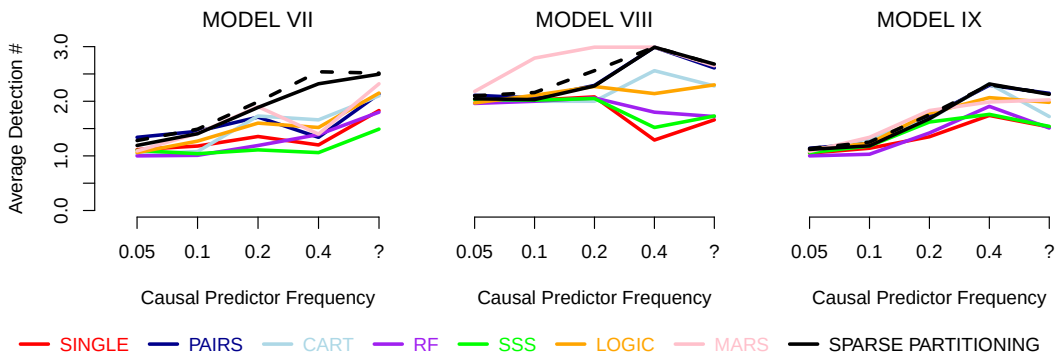


Figure 4.6: *Results of Study Six. Binary response values and 2, 3 or 4 causal predictors. In addition to the standard 8 methods, the black dashed line indicates the performance of* Sparse Partitioning *using a probit link function.*

4.1.9 Study Seven: Correlated Predictors

To generate datasets displaying realistic patterns of linkage disequilibrium, I used the program `ms` (Hudson, 2002), which is provided on its author's website, hosted by the University of Chicago. The command `ms 1000 1 -s 20 -r 10 20 -F 100` will simulate 1000 individuals typed wildtype or mutant for 20 SNPs. I concatenated the results of 100 such runs, with every second SNP removed, to obtain a dataset of 1000 individuals typed for 1000 SNPs. To give an indication of the extent of LD this generated, if I were to filter the dataset so that no pair of predictors remained with a squared correlation greater than 0.8, approximately half the predictors would be removed. Again, three different underlying relationships were examined, similar in nature to those in Study One, except this time there were four causal predictors.

Model	Underlying Relationship
X	$Y = aX_1 + bX_2 + cX_3 + dX_4$
XI	$Y = f_1(X_1, X_2) + f_2(X_3, X_4)$ where f_1 maps to $\{0, a, a, b\}$ and f_2 maps to $\{0, 0, 0, c\}$
XII	$Y = f_1(X_1, X_2) + f_2(X_3, X_4)$ where f_1 maps to $\{0, a, b, c\}$ and f_2 maps to $\{0, d, e, f\}$

The coefficients a, b, c, d, e and f were sampled at random from a standard normal distribution. Unlike the other studies, in this one I used a prospective method of sampling, first picking from the dataset four predictors, then from these generating a response. By simulating in this direction, the variance of the response values will be much more limited, heavily focused on the underlying relationship value corresponding to the most likely state of the causal predictors. To counteract this, having generated response values for 1000 individuals, I pick 100 displaying a broad spectrum of values. On very rare occasions, the response values lacked

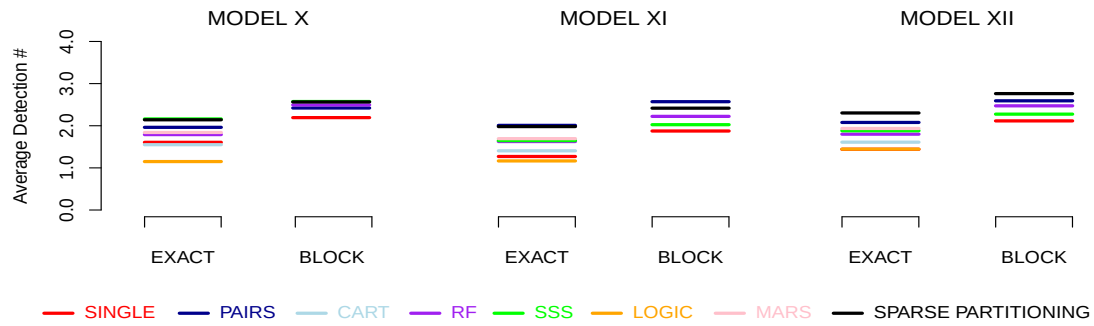


Figure 4.7: *Results of Study Seven. To mimic the patterns of linkage disequilibrium observed in fine-scale association studies, the predictors were generated so that strong pairwise correlations were present within each block of ten successive predictors. As the datasets were generated prospectively, it was not possible to dictate the causal predictor frequencies. For this study only, two alternative scoring methods were used, EXACT and BLOCK, details of which are provided in the main text.*

variation to such an extent that this was not possible. For these cases, I discarded the four predictors and reselected new ones. Because of the prospective sampling, it was not possible to fix the causal predictor frequencies.

In this study, I experimented with two scoring systems. The first, EXACT, identical to that used in the other studies, made no allowance for correlations. This could be considered overly harsh. Suppose a method identified as associated a predictor near to, and so in high correlation with, a causal predictor. Strictly speaking, this would be a false positive and score zero, even though its detection would still be a helpful indicator of the region likely to contain the true association. Therefore, the second system, BLOCK, scored each block of ten predictors. For the method *Single*, which considers one predictor at a time, so makes no allowance for LD, blocks were scored according to their best scoring predictor. For *Pairs*, *RF*, *SSS* and *Sparse Partitioning*, blocks were scored by summing over their ten predictors. *CART*, *Logic* and *MARS* only return precise models, rather than weightings for each predictor, so could not be scored with this system.

Figure 4.7 provides the results for this study. As is almost necessarily the case, the detection accuracy improved using the second scoring system, and overall *Single* and *RF* benefited most from the change. In most cases, however, the ranking of the five methods scored under both systems was preserved. Once more, *Sparse Partitioning* performed admirably, coming top for five scenarios, beaten only by *Pairs* in the sixth.

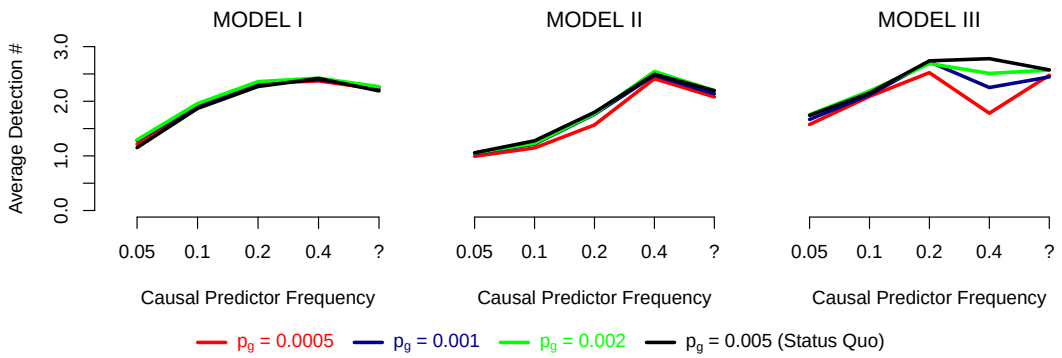


Figure 4.8: *Results of Study Eight. Each line corresponds to running* Sparse Partitioning *with a different prior probability of association.*

4.1.10 Study Eight: Effect of Prior Choice

The most important input setting for *Sparse Partitioning* is p_g, the prior probability of association for each predictor. The other parameters, such as maximum number and size of groups, or the variance of the coefficient priors, can almost always be left at their default values. In this study, I investigated the effect of different values for p_g, as opposed to keeping it at $5/N = 0.005$, the *status quo* for other studies. Datasets were generated using the same three underlying relationships of Study One.

Figure 4.8 presents the results for four choices: $p_g = 0.0005$, 0.001, 0.002 and 0.005. For the first two models, the difference in performance was slight, but as expected the latter two choices, which are closest to the true case, have performed best. The difference was more noticeable for higher causal predictor frequencies under Model III. The results suggest it is advisable to verge on the cautious side when setting p_g, which agrees with the general message that less restrictive is better.

4.1.11 Study Nine: Examine Effect of Number of Iterations

Naturally, the more iterations *Sparse Partitioning* can be afforded, the better its performance. In general, I ran the method for 200 iterations, as this allowed datasets to be analysed in under a minute. To compare this number to other methods, which often sample for upwards of 100,000 iterations, it is worth remembering that *Sparse Partitioning* performs approximately $2N$ samplings per iteration. In this study, I generated datasets using the same approach and models of Study One. To each dataset, I applied *Sparse Partitioning* four times, varying the number of iteration from 100 to 800. Figure 4.9 presents the results. As with many of the studies, the differences in performance showed up more as the models became more complicated. This study suggests that, while 200 iterations (the blue line) seems fairly sufficient,

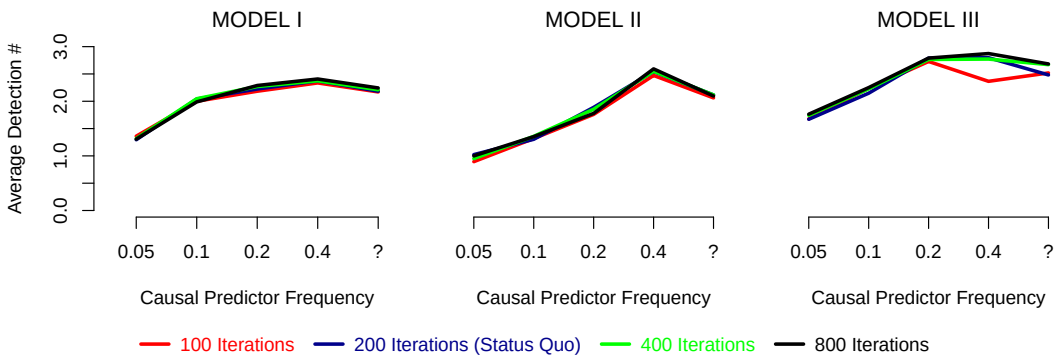


Figure 4.9: *Results of Study Nine. Each line corresponds to running* Sparse Partitioning *for a particular number of iterations.*

slightly better performance might have been obtained by using more.

Incidentally, for both Studies Nine and Ten, I generated datasets from scratch, rather than reusing those generated in Study One. Therefore, the concordance between runs of *Sparse Partitioning* across these three studies suggests that 100 datasets was sufficient to provide a reasonable measure of detection score.

4.1.12 Study Ten: Non-Disjoint Underlying Relationship

It is conceivable that a predictor might feature more than once in the underlying relationship. This could, for example, correspond to a genetic variant involved in two or more pathways. This study considered three models where this was the case.

Model	Underlying Relationship
XIII	$Y = X_1 \times X_2 + X_2 \times X_3$
XIV	$Y = f_1(X_1, X_2) + X_2 \wedge X_3 + 2X_4$ where f_1 maps to $\{0, 1, 2, -1\}$
XV	$Y = f_1(X_1, X_2) + 2X_2 \oplus X_3$ where f_1 maps to $\{0, 1, 2, -1\}$

Although *Sparse Partitioning* has generally performed best in this study, simulation under Model XIII reveals a possible shortcoming of the sparsity assumption. Suppose the underlying response value is increased by a multiplicative interaction between a pair of low frequency predictors. The two most likely predictor states will be $(0, 0)$ (generally corresponding to low response samples) and $(1, 1)$ (for high response samples). By contrast, the states $(0, 1)$ and $(1, 0)$ will be very unlikely, as they effect the same underlying response value as $(0, 0)$, but have a much lower chance of occurring due to the low predictor frequencies. When confronted with such data, the method *Single*, which considers predictors independently, will have a

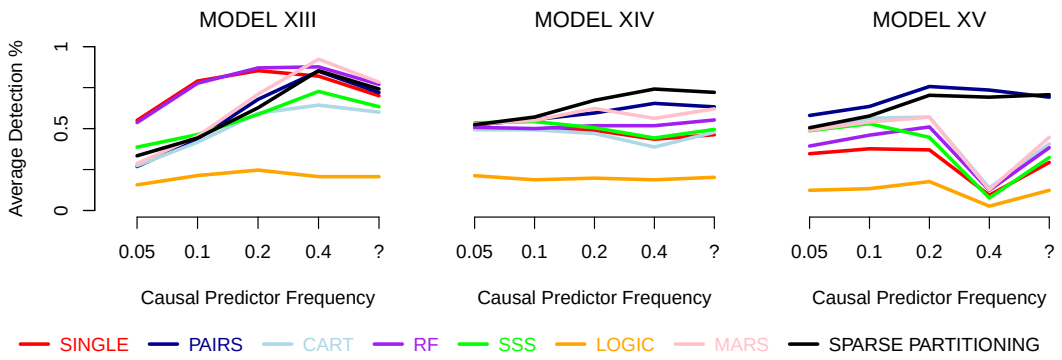


Figure 4.10: *Results of Study Ten. In each underlying relationship, one predictor contributed twice, so that the true partitioning was no longer disjoint. In this study,* Sparse Partitioning *was outperformed under Model XIII for low causal predictor frequencies. The main text suggests reasons why this might have occurred.*

large advantage, as both predictors should have strong marginal effects. Conversely, *Sparse Partitioning* would be expected to perform badly, as the improvement in fit of deducing the true multiplicative interaction is unlikely to be large enough to offset the penalty of including an extra predictor. The hope is, however, that the prior belief in the rarity of such a situation is correct. Furthermore, as *Single* is built into the implementation of *Sparse Partitioning*, in this scenario the causal predictors would still be discovered.

4.2 Real Datasets

In this section, I consider three previously analysed association studies. I am indebted to the individuals who made each available. The first dataset is taken from the "2010 Project" (http://walnut.usc.edu/2010), a large-scale study of the plant *Arabidopsis thaliana*, which I obtained from Keyan Zhao and Bjarni Vilhjálmsson of Magnus Nordborg's Lab. The second dataset is part of the International HapMap Project (http://hapmap.ncbi.nlm.nih.gov), given to me by Antigone Dimas while she was part of Emmanouil Dermitzakis' Group. The third dataset looks at mice and was provided by Jon Krohn in association with William Valdar and Jonathan Flint.

In all cases, the datasets have previously been analysed by their respective groups, so I was keen to see what additional insights *Sparse Partitioning* might offer. Except where stated, I ran *Sparse Partitioning* with $r = 1$ and other parameters at their default values, the most important of which being $K = 4$, $S = 4$, $C = 3$ and $p_g = 5/N$.

4.2.1 2010 Project: Pilot Data

The project's pilot dataset examines 95 accessions of *Arabidopsis thaliana*, each measured for ten phenotypic traits. As a straightforward demonstration of my method, I focus mainly on the tenth phenotype, expression levels of the `FRIGIDA` gene.

As explained in Nordborg *et al.* (2005), this study examined "pairs of individuals from 25 local 'populations' (typically sampled within a few hundred meters of each other, often much closer)" together with "a worldwide survey of commonly used stock centre accessions". Geography plays an important role in this experiment, with samples selected from countries across Europe, as well as North America and Asia.

Arabidopsis exist largely as collections of naturally occurring inbred accessions, so it proves sufficient to genotype only one sample per accession. By contrast, it is prudent to phenotype many members in order to reduce environmental and experimental noise. The manner in which accessions were sampled resulted in very high levels of relatedness between accessions from neighbouring fields. Combined with this, geographical effects can be striking, because accessions will tend to be influenced by and adapted to their surroundings. For example, those found in bleak climates such as Northern Scandinavia showed marked differences in time until flowering to those from more sunny locations.

The Nordborg group published two closely related papers focused on this dataset. The first (Aranzana *et al.*, 2005) demonstrated that, even in spite of the presence of major confounding factors, it was possible to identify four known pathways simply using naïve one-predictor-at-a-time tests. The second paper (Zhao *et al.*, 2007) applied a more sophisticated method, one which took confounding into account, similar to the way I shortly describe. Although both papers primarily searched for association with haplotype, which typically represent 500-600 bp fragments of DNA, I was provided with, and used instead, the 5,419 raw SNP genotypes.

Estimation of Relatedness and Population Structure

For a pair of individuals, the "coefficient of inbreeding" (Wright, 1922) equals the probability that, on an autosomal (non-sex) chromosome, a locus of one individual is "identical by descent" with that of another. For example, consider identical twins. If we examine a particular locus for each, there is a 25% chance both come from the mother and a 25% chance both come from the father. Their coefficient of inbreeding would therefore be 1/2. By similar logic, this coefficient can be deduced for parent-child pairs (1/4), full siblings (1/4), half siblings (1/8), and so on. These values correspond to the "kinship" matrix $\boldsymbol{K}$ (size $n \times n$), where $K_{ii'}$ is the coefficient of inbreeding between individuals i and i'. The "coefficient of relatedness" is defined as twice the coefficient of inbreeding, and so is a measure of closeness between 0 and 1.

The standard way to estimate relatedness is to consider observed "identity by state" (Astle and Balding, 2009). This occurs when two alleles are the same, whether or not their heritage can be traced back to a (recent) common ancestor. I based my method on that suggested by Zhao *et al.* (2007), for each pair of accessions calculating $K_{ii'} = \sum_g \mathbf{1}(X_{ig} = X_{i'g})/N$, a measure between 0 and 1 of how similar two individuals' genotypes. I then standardised these values with the transformation

$$K_{ii'} \to 0.5 \times (K_{ii'} - \bar{K})/(1 - \bar{K}),$$

where $\bar{K}$ is the mean value recorded across all distinct pairs $i \neq i'$. Here, a negative value implies a pair is less related than by chance, so I reset these to zero. These transformed values became the estimates of kinship.

Population stratification is typically represented by the "population structure" matrix Q. Were this known with certainty, it would have n rows and a number of columns equal to one less than the number of (founder) populations present. In this case, Q_{ij} would indicate the proportion of the ith sample's genome which originated from the jth population. In practice, the number of populations is not normally known, so estimation of Q includes determining a suitable number of columns.

I opted for a principal component based approach. First, I obtained the covariance matrix of SNPs (size $N \times N$), by centreing columns of $\boldsymbol{X}$, then calculating $\boldsymbol{X}^T\boldsymbol{X}/n$. Each element of this matrix reflects the similarity between a pair of predictors. I then used eigenvalue decomposition to obtain the top eigenvectors $\boldsymbol{e}_j$ (length N). Each of these should hopefully indicate the collection of SNPs which best distinguish geographic differences between accessions. Finally, I found the projection of the predictors onto these vectors, producing new vectors $\boldsymbol{X}\boldsymbol{e}_j$ (length n), which became the columns of Q. As I describe shortly, I decided how many vectors to use by studying quantile-quantile plots.

This approach is very similar to, and was inspired by, the `EIGENSTRAT` algorithm of Patterson *et al.* (2006). The key difference is that they find the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and immediately employ these as columns of Q. There are at least two reasons for preferring their choice. Typically, the dimension of $\boldsymbol{X}\boldsymbol{X}^T$ is far less (size $n \times n$) than that of $\boldsymbol{X}^T\boldsymbol{X}$ (size $N \times N$) so the decomposition is much faster. Secondly, by investigating the distribution of the top eigenvalues, a connected paper by Price *et al.* (2006) provides a theoretical basis for deciding the correct number of eigenvectors (columns of Q) to use. I chose my approach because I felt better able to understand the logic behind decomposing $\boldsymbol{X}^T\boldsymbol{X}$. The resulting eigenvectors provide the most variable projections of the data. If we are to suppose the effects

of "genetic" confounding, among which population structure is included, greatly overshadow any true signal present, then these effects should be picked up by the top projections. In any case, the choice of decomposition turns out to be of little importance. The two sets of eigenvectors are intimately linked by the singular value decomposition of $\boldsymbol{X}$ (Golub and Reinsch, 1970) and, as a result, produce very similar estimates for Q.

The major rival to `EIGENSTRAT` is `STRUCTURE` (Pritchard *et al.*, 2000), which I discussed earlier. The latter is certainly more sophisticated, incorporating elements of advanced coalescent theory. However, I considered principal component analysis more applicable for large datasets, easier to include in the implementation and I liked the fact that it is not limited to detecting population structure; it could, in principle, account for other types of genetic noise. Reassuringly, Engelhardt and Stephens (2010) have recently demonstrated the similarity between the two main approaches, suggesting the final choice is of little significance.

Correcting for Relatedness and Population Structure

The favoured approach of the Nordborg Lab is to use a mixed model (Yu *et al.*, 2006), which takes the form

$$\boldsymbol{Y} = f(\boldsymbol{X}) + Q\boldsymbol{d} + \boldsymbol{u},$$

where $\boldsymbol{u}$ is a random variable with distribution $\mathbb{N}(0, 2\sigma_g^2\boldsymbol{K})$. This reflects the fact that for two individuals that are very close, we expect similar contributions due to relatedness. The variable σ_g^2 allows flexibility in the component of variation attributable to this effect. Typically, this model is applied one-SNP-at-a-time by setting $f(\boldsymbol{X}) = f(X_g)$, therefore producing estimates of $\boldsymbol{d}$, $\boldsymbol{u}$ and σ_g^2 for each predictor (e.g. Kang *et al.*, 2010).

A similar set-up is readily adopted by *Sparse Partitioning*. Consider the Cholesky decomposition $\boldsymbol{K} = \boldsymbol{E}\boldsymbol{E}^T$. If the random variable $\boldsymbol{u}'$ is distributed $\mathbb{N}(0, \boldsymbol{I}_n)$, then the transformation $\boldsymbol{u} = \boldsymbol{E}\boldsymbol{u}'$ will have distribution $\mathbb{N}(0, \boldsymbol{E}\boldsymbol{I}_n\boldsymbol{E}^T) = \mathbb{N}(0, \boldsymbol{K})$. Therefore, to incorporate a mixed-model in the analysis, I simply include Q and $\boldsymbol{E}$ within the confounding matrix: $\Omega = [Q\ \boldsymbol{E}]$. A possible downside is that *Sparse Partitioning* insists that the confounding variance equals a constant factor of the residual variance. Therefore, σ_g^2 is fixed to σ^2/r'. However, my feeling is that the model has sufficient flexibility elsewhere to overcome this limitation.

Figure 4.11 demonstrates some of the steps involved in estimating $\boldsymbol{K}$ and Q for the 95 *Arabidopsis* samples. The first plot represents an initial estimate of the pairwise relatedness $2\boldsymbol{K}$; darker colours indicate values closer to 1. It is just possible to make out red dots off the diagonal, indicating distinct pairs of accessions with almost identical SNP values. In total, I discovered 5 such groups (11 samples in total), leading me to remove 6 accessions.

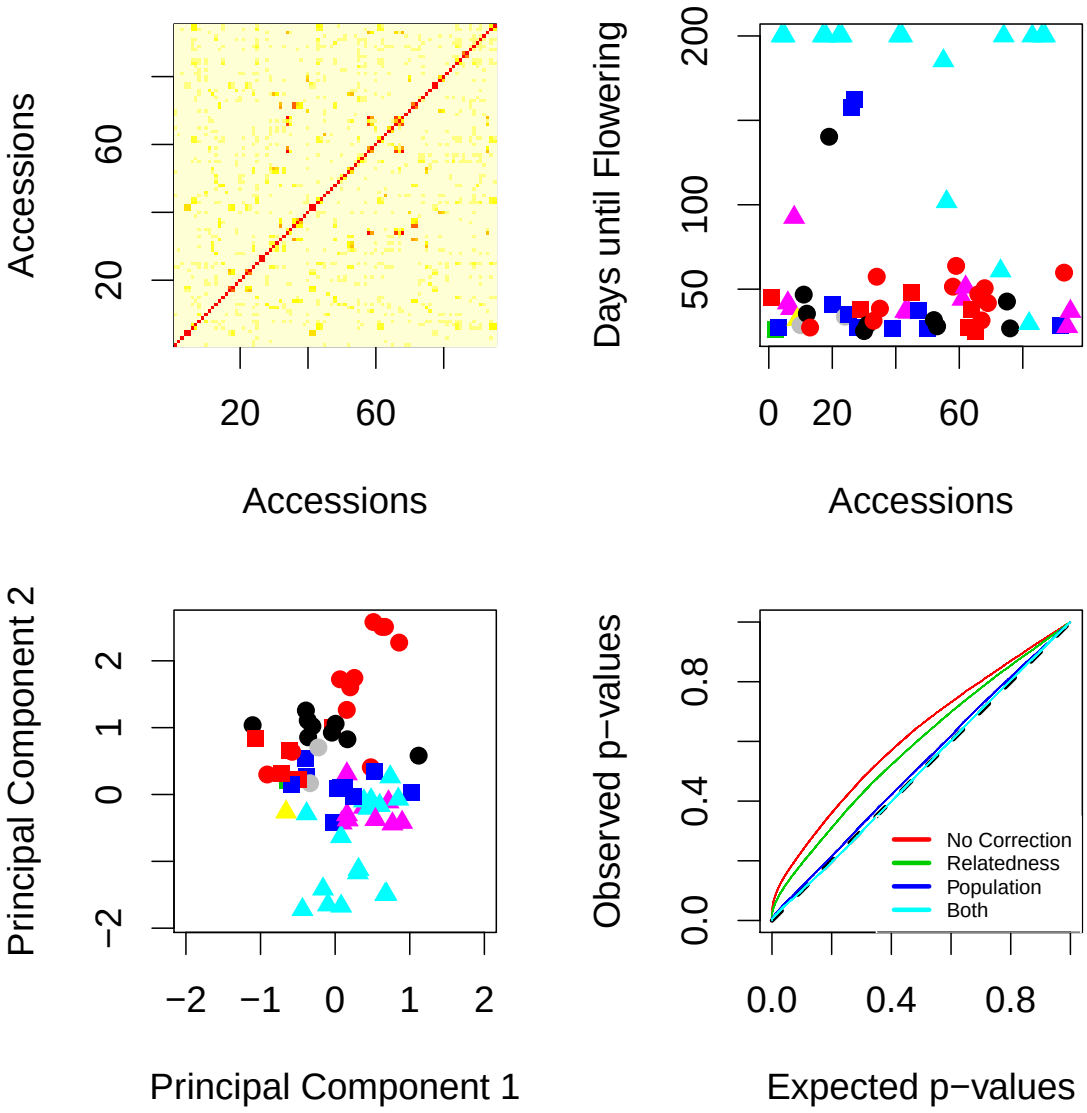


Figure 4.11: *The first plot records the relatedness between pairs of accessions. Necessarily symmetrical, because $K_{ii'} = K_{i'i}$, it is possible to make out a few isolated dots off the diagonal, indicating distinct samples which are very highly related. The middle two plots demonstrate the severity of population structural effects. Each plot shape and colour identifies a particular location of origin. The Swedish accessions (light blue triangles) have markedly different measurements for the phenotype which counted days until flowering, while the first two principal component axes, derived only from genotype data, well separate these samples from the rest. The final plot demonstrates the strategy I used when correcting for these two types of confounding. The red line shows the p-values obtained by the method* Single *when regressing the raw phenotypic values on the SNP data. This is heavily biased towards small values, as indicated by its position above the diagonal. By considering the spread of p-values obtained after correcting for relatedness (green), population (dark blue) and both together (light blue), it was possible to gauge which were needed and, in the case of population structure, estimate the number of "population axes" required.*

The second plot reports measurements of the first phenotype, days until flowering. The shapes and colours of points distinguish the 24 different countries of origin. The measurements were truncated, so a time of 200 days indicated the accession was yet to flower when the experiment ended, and had most likely died. It is noticeable that Swedish samples (light blue triangles) account for all of the truncated measurements, supporting the notion that geography affects certain traits. The third plot shows how well the geography could be inferred from the principal components. With no knowledge of the data other than the SNP values, it was easily possible to cluster the main locations of accessions. Additionally, two samples recorded extreme values, comfortably off the plotted scale, so I removed these from subsequent analysis.

The fourth plot demonstrates my method for determining how many principal axes to include in Q and a suitable value for r'. The red line plots the p-values obtained by applying *Single* to all 10 phenotypes (54,190 tests in total). This shows the extent to which confounding affected the study. Under an assumption that the majority of associations were spurious, we would expect corrected p-values to be uniformly distributed on (0,1) and so produce a line coincident with the diagonal. However, in this case, 16% of predictor-response pairs were significant at a nominal 5% level. The green, dark blue and light blue lines show the effects of supplying *Single* with only $\boldsymbol{K}$, only Q or both $\boldsymbol{K}$ and Q. I picked the number of population axes (5) and confounding variance ($r' = 2$) by experimenting with values until I was satisfied with the fit. Supposing sensitivity is more valued than specificity, it is probably best to err on the conservative side; generally, we would prefer to remove too little confounding than risk removing too much true signal.

Results

Having discarded 6 samples on account of excessive relatedness and 2 due to extreme principal component values, I began analysis with a reduced list of 87 accessions. The tenth phenotype recorded expression levels for the `frigida` gene, a continuous non-negative measurement. Therefore, I first applied a logarithmic transformation so that the values more closely resembled draws from a normal distribution. I set *Sparse Partitioning* to filter predictors using an r^2 threshold of 0.8. This reduced the total number of SNPs to 3,289. In total 50,428 (10.7%) predictor values were missing, but for this run I chose not to impute.

The top two plots of Figure 4.12 compare the p-values obtained from *Single* to the posterior probabilities of association of *Sparse Partitioning*. My method identified just one strong association, coinciding with the third strongest hit of *Single* and suggesting that, in this case, the simple underlying relationship of the one-predictor-at-a-time method might be appropriate. For both methods, the strong associations lay very close to the `FRIGIDA` region, marked

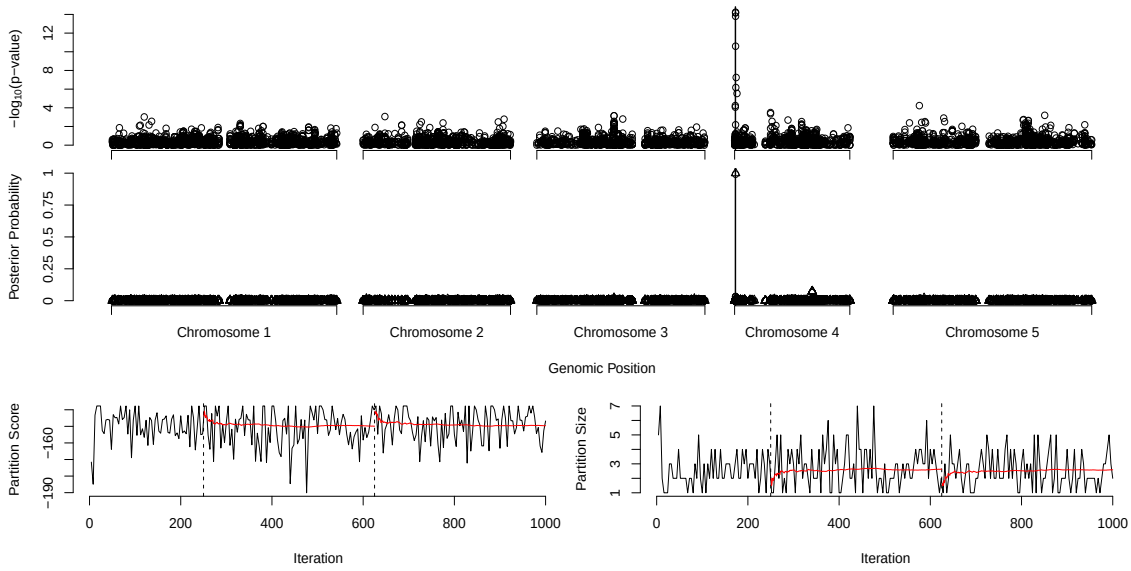


Figure 4.12: *Analysis of expression levels of* `FRIGIDA` *for* Arabidopsis thaliana. *The top plot shows p-values obtained by* Single. *The middle plot shows posterior probabilities of association from* Sparse Partitioning*; the results of two runs are plotted, but the separation (circles vs triangles) is barely visible. The bottom row provides trace plots for the partition score and size. The vertical dashed lines mark the two sections of iterations used to produce estimates, while the red lines indicate the running means.*

by a solid vertical line, providing further evidence that the results were accurate.

A possible concern with *Sparse Partitioning* is that its generality might lead to overfitting on occasions when simpler models are more appropriate. Here, with *Sparse Partitioning* declaring only one strong association, this does not appear to be the case. The bottom two plots track the posterior score and size of the current partition at each stage of the Markov chain, with red lines indicating running means for the two halves of the kept iterations. It is interesting to note that, although *Sparse Partitioning* clearly returned just one association, the average partition size was approximately 2.5, so while overfitting occurred locally, it did not affect the final results.

I repeated the analysis using imputed data provided by the algorithm `fastPHASE` (Scheet and Stephens, 2006), allowing me to compare the prediction accuracy of each method via LOOCV. The linear model containing only the top hit from *Single* explained 44% of the variance, agreeing closely with *Sparse Partitioning*'s estimates of 38% or 42% explained when using the raw or imputed data. Once again, even though overfitting appeared to occur at a per-state level, the method seemed able to produce meaningful estimates.

4.2.2 HapMap Data

The HapMap dataset consisted of 109 individuals, each typed for 1,186,075 SNPs and measured for expression levels of 2,682 genes. Dr. Antigone Dimas had previously mined the data for interactions using her own novel method (described in Dimas, 2009). Dr. Dimas was interested in searching for *cis* interactions, which she classed as interactions within 1 Mbp of the gene probe start site. First, she used the locations of known recombination hotspots to segment the genome into recombination intervals. For each gene, she then applied one-predictor-at-a-time testing for all SNPs within 1 Mbp, recording the top scoring SNP within each recombination interval. Typically, this provided her with a list of about 30 SNPs per gene, for which she then tested all pairwise interactions.

Dr. Dimas provided me with a list of the genes showing most evidence for an interaction. Taking the top four of these, I was interested in seeing how *Sparse Partitioning*'s results would compare. Figure 4.13 presents the results for `MTHFR`, the third of these genes, located approximately 11.8 Mbp along Chromosome 1. For each of the 763 SNPs in the 2 Mbp region, the top plot displays the p-value obtained by *Single*, while the middle plot reports the posterior probability of association from *Sparse Partitioning* (circles correspond to run one results, triangles to run two). The solid vertical line marks the location of the gene, while the two dashed vertical lines mark the locations of the SNPs declared interacting by Dr. Dimas. The dashed horizontal lines in the top two figures provide estimates of the 5, 25 and 50% significance thresholds for the top association of each method, calculated using permutation tests.

The top hits of *Sparse Partitioning* were SNPs rs2286139, rs2643888 and rs2279703 ("SNPs 1, 2 and 3"), with posterior probabilities of association 0.57, 0.96 and 0.96, respectively. The first two correspond to the SNPs for which Dr. Dimas found evidence of an interaction. It is no coincidence that SNPs 2 and 3 received equal probabilities. Their values matched for 106 of the 109 individuals, so SNP 3 was removed in *Sparse Partitioning*'s preprocessing step and subsequently assigned the same posterior estimates as SNP 2. The second and third SNPs ranked highly in both sets of results, comfortably exceeding the 5% permutation threshold each time. More interesting was the first SNP. In *Single* it was assigned a p-value just shy of 0.01, making it the 74th highest ranked association, so perhaps unlikely to be followed up on the strength of its marginal association alone.

Sparse Partitioning returned a posterior probability of interaction of 0.42 between SNPs 1 and 2/3 (indicated in the plots by the horizontal arrows). The question is whether this offers evidence for an interaction. At first glance, it might appear not to. This probability is less than 0.5, a threshold commonly applied to Bayesian methods as it indicates balance of probabilities. Furthermore, the plot of marginal posterior probabilities might suggest the best

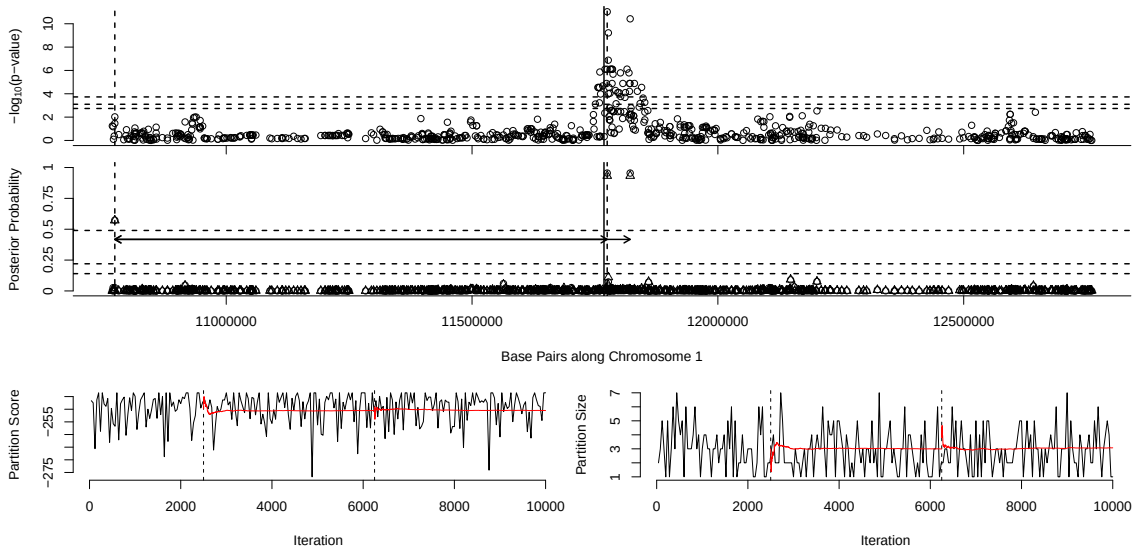


Figure 4.13: *Analysis of expression levels of* `MTHFR` *using HapMap data. The top plot shows results from* Single*, the middle plot shows results from two runs of* Sparse Partitioning *(circles correspond to run one, triangles to run two). The bottom row plots the score and size of the current partition for each iteration of the MCMC sampling.*

fitting partition declared SNPs 1 and 2 associated. If this was true, then *Sparse Partitioning* will have assigned a prior probability of 0.5 to their interaction, so a posterior probability of 0.42 would appear to provide slight evidence against them interacting.

On the other hand, this argument is to some extent discredited by noticing that the average partition size was just over three. Furthermore, the first SNP's marginal probability was 0.57, so this was automatically an upper bound for the interaction probability. Supposing the true partition does contain only SNPs 1 and 2, it is straightforward to work out the (conditional) posterior probabilities for the two possible partitions, a feature offered by the deterministic version of the method (Chapter 5). These probabilities turn out to equal 0.57, for the partition with an interaction, and 0.43, for the partition without.

This discussion highlights the difficulty of assigning significance to an interaction, a problem I touched upon earlier. As interactions are likely to be relatively rare, perhaps simply flagging possible ones is sufficient, so that then they can be investigated further. For example, Figure 4.14 looks closer at models suggested by *Sparse Partitioning*. The top row corresponds to three of the partitions that would have been examined: the partition with just SNP 1 associated, the partition with just SNP 2 associated and the partition where these two predictors interact (with degrees of freedom 3, 3 and 7, respectively). Notice how some of the cell counts are small. In particular, the second SNP takes value '2' only once. Therefore, in the bottom row, I consider models obtained by merging states '1' and '2' for each SNP, which

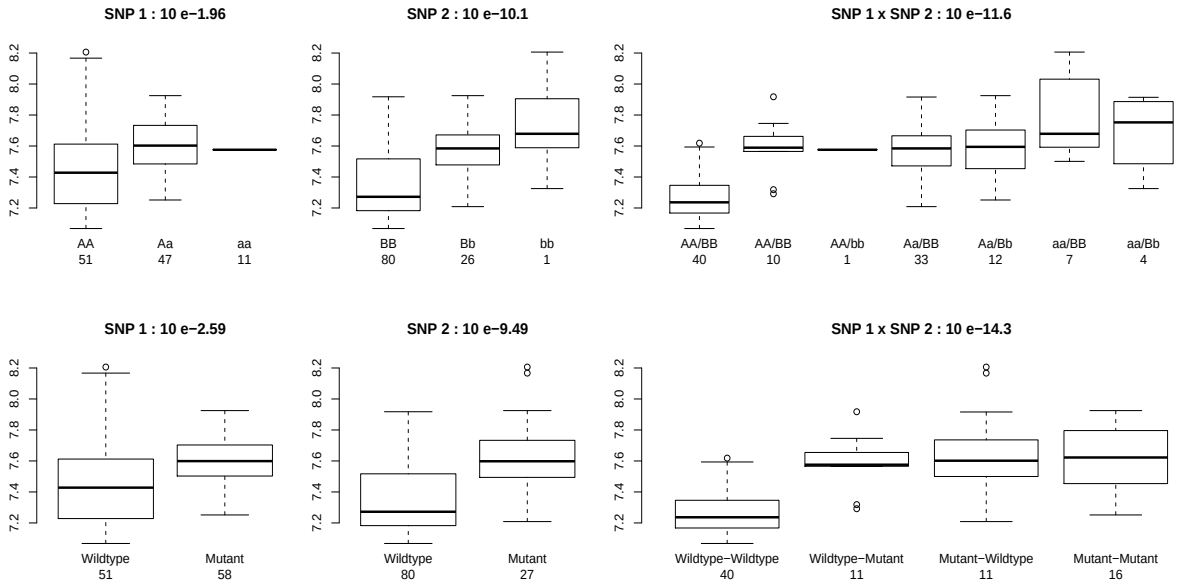


Figure 4.14: *Boxplots from the HapMap data. Each boxplot shows how gene expression values vary according to predictor states for a particular partitioning. The top row corresponds to three partitions suggested by* Sparse Partitioning*, the numbers under each cell denote the cell counts. The bottom row presents essentially the same information, except cells have been merged to avoid small counts, so producing simplified models. For each plot, the p-value indicates the strength of evidence for the model obtained via a maximum likelihood test, by comparison with a null hypothesis of no associations.*

might provide a better insight into the underlying system. The final boxplot suggests the true model might involve a threshold interaction, whereby a mutation in either SNP 1 or 2/3 triggers increased expression. I assign p-values to each model based on frequentist maximum likelihood tests; in all cases partitions are compared to the null model. The p-value for the threshold interaction model (2 degrees of freedom) is less than 10^{-14}, suggesting at the very least it should be considered further.

4.2.3 Mouse Data

Jon Krohn generously provided me with CD4 counts for 1,274 "heterogeneous stock" mice (Solberg *et al.*, 2006), with genotypic values for 770 SNPs along the length of Chromosome 5. Krohn had previously analysed this data using `Bagphenotype`, software designed by Dr. William Valdar (`www.unc.edu/~wvaldar/bagphenotype.html`). He chose this chromosome as his analysis had suggested a possible interaction with sex at one of the loci. The response values were continuous, while the predictors were tertiary. Only a tiny proportion of genotypes (0.1%) were missing, so I saw no need to impute values and instead left them as unobserved. In addition, I was provided with the gender of each mouse, which I coded as a binary variable and included in the set of predictors.

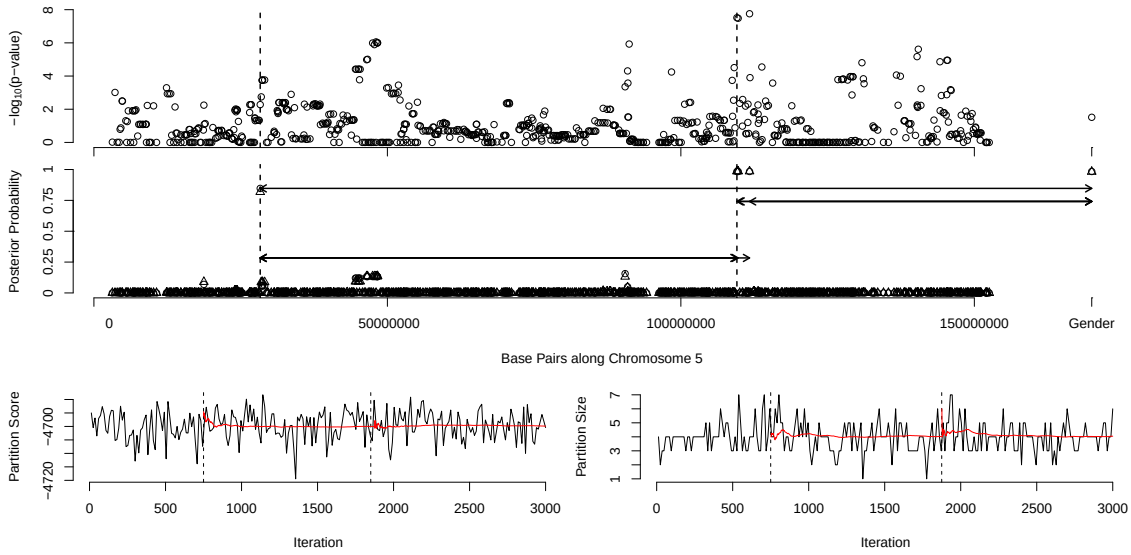


Figure 4.15: *Analysis of CD4 count in mice. The top row reports p-values obtained from* Single*, the middle row shows posterior probabilities from two runs of* Sparse Partitioning *(horizontal arrows indicate pairwise probabilities of interactions), the bottom row provides trace plots for the score and size of partitions encountered at each step of the MCMC sampling.*

As the chromosomal region was a subsection of a genome wide study, I decided a prior probability of association of 10^{-4} was appropriate for each SNP. There is strong prior knowledge that CD4 counts are linked to gender (e.g. Maini *et al.*, 1996), so I decided upon a prior probability of 0.5. As Figure 4.15 demonstrates, the top hits from *Single*, SNPs CEL-5_106584673 and rs13478460 ("SNPs 2 and 3"), which due to linkage disequilibrium are almost identical, persisted in *Sparse Partitioning*. In addition, my method declared associated SNP rs13478156 ("SNP 1"). As indicated by the horizontal arrows, *Sparse Partitioning* found evidence of interactions between gender and SNPs 1 and 2/3. To test the effect of prior choice, I repeated the analysis with prior probabilities $\{10^{-4},0.1\}$, $\{10^{-3},0.5\}$ and $\{10^{-3},0.1\}$ for each SNP and gender, and obtained very similar results on each occasion (results not shown).

In this example, the ability to consider multiple copies of predictors came in very useful. If predictors could only contribute once, *Sparse Partitioning* would have been unable to simultaneously consider the three pairwise interactions involving SNPs 1, 2/3 and gender. The boxplots in Figure 4.16 provide an insight into the way predictors appear to interact. Again, all p-values are obtained from performing marginal likelihood tests against the null model. While SNP 2 demonstrates clear marginal effects (p-value of $10^{-7.5}$), the individual contributions of SNP 1 and gender are only moderate ($10^{-1.5}$ and $10^{-2.3}$). The strongest pairwise interaction is between SNP 2 and gender ($10^{-10.5}$), although this represents only a modest improvement on the p-values of two marginal models. The corresponding boxplot (top right) suggests the SNP's effect is threshold in females and additive in males. By contrast, combining SNP 1 and

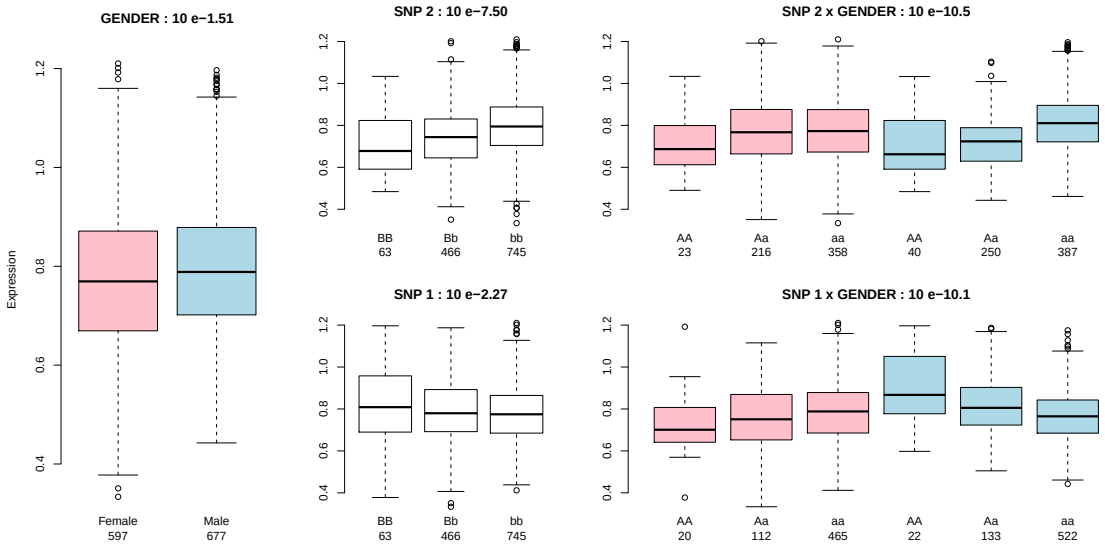


Figure 4.16: *Boxplots from the mouse data. Each boxplot shows how CD4 count varies according to predictor state for different models. The left three plots consider marginal models, the right two plots consider interactions. Pink boxes correspond to female mice, while blue boxes correspond to males. The figures beneath cells indicate cell counts, while p-values for each model are obtained from maximum likelihood tests by comparing with the null model of no associations.*

gender has a more marked effect (bottom right), resulting in a p-value whose order is 6 higher than the sum of the orders of the p-values for the two marginal models. Furthermore, there seems to be a defined difference between the effect of SNP 1 in females and males, with the trend switching directions between the two sexes.

Chapter 5

Deterministic Sparse Partitioning

The main drawback of Sparse Partitioning *is its scalability. While the MCMC sampling is asymptotically valid for any size of dataset, the number of iterations required for reasonable convergence will depend on the nature of the experiment. In this chapter, I describe a deterministic version of the method, so called because it removes the stochastic component and will return identical results each run. Although the changes have an impact on performance, they vastly increase the method's usability, as I demonstrate by applying this version to two whole genome datasets.*

5.1 Motivation

Sparse Partitioning relies on its MCMC sampling obtaining reasonable convergence. The sizes of n and N affect this process in a fairly predictable manner; many of the calculations scale linearly in n, while the number of partitions searched during each iteration is linear in N. However, the most computationally demanding element of the algorithm is solving $\boldsymbol{BA} = \boldsymbol{J}^T\boldsymbol{Y}$. The time this takes depends on D, the degrees of freedom of the model being tested, which in turn is linked to the complexity of the true underlying relationship. This makes it impossible to put an exact figure on the size of dataset *Sparse Partitioning* is realistically able to handle. When the underlying relationship is simple, for example, containing just one causal variable, *Sparse Partitioning* could easily analyse many tens of thousands of predictors. By contrast, were six or more variables to contribute towards $f(\boldsymbol{X})$, the method might struggle to analyse a few hundred predictors in the same time. Nonetheless, I can confidently state that in its current form, *Sparse Partitioning* is suitable for thousands of predictors, rather than hundreds of thousands. This automatically precludes its use on, for example, fine-scale whole genome datasets, where the number of predictors can comfortably exceed 500,000.

In addition to size constraints, *Sparse Partitioning*'s usability will suffer from a wider malaise surrounding Bayesian methods. Consider a biologist, whose first taste of statistics

will likely involve maximum likelihoods, p-values and confidence intervals. When confronted with a school of thought whose fundamental statement is that parameters, which only ever exist as fixed states, should be modelled as if they were random variables, it is understandable that they might more readily accept a frequentist alternative. This problem is only exaggerated when nonlinearity is involved, as then the Bayesian method is faced with the challenge of describing distributions over an increasingly complex model space.

These considerations led me to design a deterministic version, one which returns a definite model and where the algorithm is not dependent on a random seed. Essentially, the choice I made was to replace the MCMC exploration by a "hill-climbing" search, one which stops as soon as a higher scoring partition can not be found. Although accuracy is reduced, the result is a version which provides more readily interpretable results and is able to reduce computation time from a number of hours to a matter of minutes.

5.1.1 Dangers of a Deterministic Search

My first attempt at a deterministic version made only a slight change to the steps of Sampling Stages One and Two. In Stage One, when considering changing the group to which predictor g belongs, having explored all possible values for I_g, I proposed, and automatically moved to, the one producing the highest scoring partition. Similarly, for Stage Two, I explored all possible values for G_{kj}, then picked the one resulting in the greatest improvement. Immediately, this approach led to inconsistencies. For example, both the choice of random seed and relabelling the predictors would change the order in which components of $\boldsymbol{I}$ are sampled and potentially affect the direction of the chain. Therefore, an early fix was to merge the N steps of Stage One with all possible Stage Two moves, and thus consider the entire neighbourhood reached by any single change to an element of $\boldsymbol{I}$ or any component of $\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_K$.

Testing suggested that this approach was unsuccessful. In its original form, *Sparse Partitioning* benefits from drawing inferences across a number of partitions. The states visited by the Markov chain will be among those with highest posterior mass, so in effect the method averages over the most probable models. By contrast, the deterministic version is a mode seeking approach, which concentrates only on detecting the single highest scoring state. When the posterior weight is highly concentrated on a single partition, both strategies have the potential to fare equally well; when the posterior weight is more diffuse, a model averaging approach will better describe the posterior distribution. A particular concern arises when there are two or more well-matched partitions within a high scoring equivalence class. If our primary goal is to detect which predictors are associated, it will become counter-productive when two closely matched partitions compete over a single equivalence class' posterior weight.

Furthermore, a hill-climbing search through partitions can be very restrictive. Suppose the true partition is $\{1\}\{2,3\}$, so involves an interaction of predictors 2 and 3 alongside an additive contribution from predictor 1. For this partition to be considered, the current state, which is forced to follow a path of increasing score, must get within one move of this model. Yet two partitions can be very similar and yet more than one move apart. In this example, even if the algorithm reaches the partition $\{1,2\}$, this is still (at least) two moves away from $\{1\}\{2,3\}$.

Based on these realisations, I decided to concentrate on exploring the space of equivalence classes $[\boldsymbol{I}]$, rather than the space of partitions $\mathbb{G}$, and adapted the search algorithm accordingly.

5.2 Deterministic Sparse Partitioning

I explain the methodology in terms of a Markov chain, but bear in mind that its stationary distribution no longer matches the posterior. Additionally, this version is only suitable for an identity or logit link function (Cases 1 or 2), for reasons which will become clear.

Sparse Partitioning concentrates on exploring the space of partitions. By contrast, the deterministic version, which I refer to as *Deterministic SP*, explores the space of equivalence classes. As a reminder, the equivalence class $[\boldsymbol{I}]$ consists of all partitions corresponding to a particular set of associations $S_{\boldsymbol{I}}$. All other aspects of the set-up, in particular the underlying relationship and choice of priors, remain the same. When it comes to moving through the model space, the algorithm is required to calculate a score for each equivalence class examined, which is proportional to its posterior probability. This score becomes

$$\text{Score}([\boldsymbol{I}]) := \sum_{\mathbb{G}\in[\boldsymbol{I}]} \text{Score}(\mathbb{G}) = \sum_{\mathbb{G}\in[\boldsymbol{I}]} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X},\mathbb{G}) \times \mathbb{P}(\mathbb{G}).$$

Current Neighbourhood $[\boldsymbol{I}]^\dagger$

In each iteration of the model search, *Deterministic SP* examines the neighbourhood $[\boldsymbol{I}]^\dagger$ containing all the states obtainable by a single change to the current model. The three possible change types involve removing a predictor from $S_{\boldsymbol{I}}$, adding a predictor to $S_{\boldsymbol{I}}$, or swapping a predictor in $S_{\boldsymbol{I}}$ with one not currently associated. The size of this neighbourhood will be $s+(N-s)+s(N-s)$, where s is the current number of predictors associated: $s = |S_{\boldsymbol{I}}|$. To calculate the score of each equivalence class requires the scoring of $B(s-1,K,S)$, $B(s+1,K,S)$ or $B(s,K,S)$ partitions, depending on the move type.

Having scored each model in the neighbourhood, the algorithm moves to the one with highest posterior weighting, provided its score exceeds that of the current equivalence class. The search is necessarily aperiodic and, being conducted over a finite space, a (local) mode

will always be reached. For particularly large or complicated datasets, there is a chance the number of iterations will becomes excessive, so a limit on the number of moves is imposed (which is adjustable by the user).

This search requires a single score for each equivalence class. For this reason, it is not possible to use a probit link function combined with latent variables (Case 3), as the method is unable to integrate over all values of $\boldsymbol{Z}$. This should not prove too much of a set-back. The choice to analyse a binary response using a probit link function was introduced in order to speed up the MCMC sampling process; but with the deterministic version, speed is no longer an issue.

Once again, the introduction of multiple copies of predictors has negligible effect on the computation time. While the size of the neighbourhood increases with C, there is no need to consider adding, removing or swapping more than one copy of each predictor, as each will result in the same model score.

5.2.1 Outputs

Deterministic SP stops when it is no longer able to improve the current model. It then bases all inferences on the final equivalence class, $[\hat{\boldsymbol{I}}]$. The method's primary outputs are $\hat{S}_{\boldsymbol{I}}$, the set of predictors declared associated by $[\hat{\boldsymbol{I}}]$, and the highest scoring partitions within this class. In the MCMC version, *Sparse Partitioning* does not differentiate between different order interactions. For example, a non-null group of size three can only be inferred by the combination of pairwise interactions reported. In contrast, by returning the highest scoring members of $[\hat{\boldsymbol{I}}]$, *Deterministic SP* will give a fuller insight into the true configuration.

To assign a level of significance to results, *Deterministic SP* calculates two sets of Bayes factors: the first examines marginal associations, the second examines possible interactions. Firstly, for each predictor, the method performs a significance test, using $[\hat{\boldsymbol{I}}]$ as either the null or alternative hypothesis. If predictor g has been declared associated in the final model, the test compares the evidence for leaving out this predictor; if predictor g has not been declared associated, the test compares the evidence for including this predictor:

$$\mathrm{BF}_g = \begin{cases} \mathbb{P}(\boldsymbol{X}, \boldsymbol{Y}|\hat{S}_{\boldsymbol{I}})/\mathbb{P}(\boldsymbol{X}, \boldsymbol{Y}|\hat{S}_{\boldsymbol{I}}^{-g}), & \text{for } \; g \in \hat{S}_{\boldsymbol{I}}, \\ \mathbb{P}(\boldsymbol{X}, \boldsymbol{Y}|\hat{S}_{\boldsymbol{I}}^{+g})/\mathbb{P}(\boldsymbol{X}, \boldsymbol{Y}|\hat{S}_{\boldsymbol{I}}), & \text{for } \; g \notin \hat{S}_{\boldsymbol{I}}, \end{cases}$$

where $\hat{S}_{\boldsymbol{I}}^{-g}$ and $\hat{S}_{\boldsymbol{I}}^{+g}$ are the sets of associations with predictor g removed or added. Posterior odds can be obtained by multiplying each Bayes factor by $p_g/(1-p_g)$, which provide an estimate of the posterior probabilities of association for each predictor. Note that a probability

of 0.5, equivalent to a posterior odds ratio of 1, serves as the threshold for determining association; the predictors which receive a probability estimate greater than 0.5 are those which will have been declared associated, and vice versa.

Secondly, for each pair of predictors in $S_{\boldsymbol{I}}$, *Deterministic SP* calculates a Bayes factor considering the extent of evidence for or against their interaction. I mentioned this idea briefly when discussing ways to gauge the strength of interactions in Chapter 3. Let $[\hat{\boldsymbol{I}}]_{gg'} \subset [\hat{\boldsymbol{I}}]$ denote the subset of partitions in which predictors g and g' interact: $\boldsymbol{I} \in [\hat{\boldsymbol{I}}]_{gg'} \Rightarrow I_g = I_{g'}$. Let $[\hat{\boldsymbol{I}}]^C_{gg'} = [\hat{\boldsymbol{I}}] \setminus [\hat{\boldsymbol{I}}]_{gg'}$ denote its complement: $\boldsymbol{I} \in [\hat{\boldsymbol{I}}]^C_{gg'} \Rightarrow I_g \neq I_{g'}$.

$$\begin{aligned}
\text{BF}_{gg'} &= \frac{\mathbb{P}(\boldsymbol{X}, \boldsymbol{Y} | [\hat{\boldsymbol{I}}]_{gg'})}{\mathbb{P}(\boldsymbol{X}, \boldsymbol{Y} | [\hat{\boldsymbol{I}}]^C_{gg'})} \\
&= \frac{\mathbb{P}([\hat{\boldsymbol{I}}]_{gg'} | \boldsymbol{X}, \boldsymbol{Y})}{\mathbb{P}([\hat{\boldsymbol{I}}]^C_{gg'} | \boldsymbol{X}, \boldsymbol{Y})} \Big/ \frac{\mathbb{P}([\hat{\boldsymbol{I}}]_{gg'})}{\mathbb{P}([\hat{\boldsymbol{I}}]^C_{gg'})} \\
&= \frac{\sum_{\mathbb{G} \in [\hat{\boldsymbol{I}}]_{gg'}} \text{Score}(\mathbb{G})}{\sum_{\mathbb{G} \in [\hat{\boldsymbol{I}}]^C_{gg'}} \text{Score}(\mathbb{G})} \Big/ \frac{|[\hat{\boldsymbol{I}}]_{gg'}|}{|[\hat{\boldsymbol{I}}]^C_{gg'}|}.
\end{aligned}$$

The second fraction adjusts for the imbalance between the numbers of partitions with and without the interaction. I believe, in this case, the Bayes factor is more informative than the posterior odds as it offsets my choice of prior. When explaining the partition prior, I discussed that more tailored weightings might be more appropriate, but would be difficult to specify due to the individual nature of each equivalence class. If that was the case, the user can at this stage utilise prior probabilities for individual pairwise interactions. For example, should they believe that, given a pair of associated predictors, there is only a 25% chance they interact, they can multiply the Bayes factor by these updated prior odds (1/3) and obtained a revised posterior probability for their interaction. It should be remembered that any revised posterior probabilities are conditional on the final equivalence class, as had the new prior odds been incorporated into the method before it began, a different final model might have been obtained.

Last of all, the method returns a measure of model fit. When the response is continuous, *Deterministic SP* calculates a posterior estimate of variance explained by averaging over all the partitions within the final equivalence class. For the binary response case, the corresponding estimate is deviance, (twice the logarithm of) the ratio of the likelihoods under the final and null models.

5.2.2 Additional Features

Most of *Sparse Partitioning*'s features carry over to *Deterministic SP*. Confounding is dealt with in an identical fashion, by the inclusion of cofactor matrices $\mathbf{\Psi}$ and $\mathbf{\Omega}$. As before, the variables in $\mathbf{\Psi}$ are able to interact with $\boldsymbol{X}$, but must also be tertiary; while those in $\mathbf{\Omega}$ can not interact with $\boldsymbol{X}$, but can take any values. It remains prudent to scan for duplicate predictors, so that these can be assigned matching posterior estimates. Otherwise, multicollinearity is not an issue. The method is interested in finding the highest scoring equivalence class, so it does not matter if high correlations lead to groups of similarly scoring models. As run time is not an issue, it is in fact better to retain these correlations, as filtering will risk discarding true signal.

Missing Data

The deterministic version must return identical results for each run. Therefore, it is not possible to resample missing data values, as doing so would introduce variation. As before, missing response values are of no importance when the aim is purely to detect association, although I mention their prediction shortly. When some predictor values are unobserved, ideally these should be imputed before analysis. If this is not possible, *Deterministic SP* is forced to take a alternative approach.

To score each equivalence class, it is necessary to obtain a marginal likelihood for each partition, even if some predictors are unobserved. We might consider replacing the marginal likelihood with $\mathbb{P}(Y^{\dagger}|\boldsymbol{X}, \mathbb{G})$, where $Y_i \in Y^{\dagger}$ only if all $X_{iS_{\boldsymbol{I}}}$ have been observed. However, this decision would create a bias towards models with more missing values; a model which results in all samples being ignored would fit the data perfectly.

Instead, for each group, *Sparse Partitioning* creates a new node corresponding to samples containing missing values for any of the predictors in that group. For example, for the case of two binary predictors, the vector $(X_g, X_{g'})$ will now have up to five nodes, the fifth one occurring when either (or both) of X_g and $X_{g'}$ are missing. Therefore, the total degrees of freedom of the linear model will be increased by up to K, and $\mathbf{\Theta}$ will be expanded accordingly. Each regression coefficient corresponding to a "missing node" is assigned the same prior as the standard coefficients, so that the marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$ can be calculated in an almost identical fashion.

This approach has the potential to introduce bias if missing values do not occur independently of the response. For example, were it the case that samples with higher response values were more likely to have missing values for a particular predictor, then the missing node might drive evidence for an association, regardless of whether this was warranted. If this was

considered a real danger, it might be necessary to impose a stricter penalty on missing values, which could be implemented by altering the prior for the corresponding regression coefficients.

Prediction of response values is straightforward. Suppose analysis of X_O and Y_O returns the equivalence class $[\hat{\boldsymbol{I}}]$. As all results are based on this final model, the method needs only calculate the expected response value for all partitions in this class, then weight according to their posterior scores:

$$\begin{aligned}\mathbb{E}(Y_U|\boldsymbol{X}, Y_O, [\hat{\boldsymbol{I}}]) &= \sum_{\mathbb{G}} \mathbb{E}(Y_U|\boldsymbol{X}, Y_O, [\hat{\boldsymbol{I}}], \mathbb{G}) \times \mathbb{P}(\mathbb{G}|\boldsymbol{X}, Y_O, [\hat{\boldsymbol{I}}]) \\ &= \sum_{\mathbb{G}\in[\hat{\boldsymbol{I}}]} \mathbb{E}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G}) \times \frac{\text{Score}(\mathbb{G}|X_O, Y_O)}{\sum_{\mathbb{G}'\in[\hat{\boldsymbol{I}}]} \text{Score}(\mathbb{G}'|X_O, Y_O)}.\end{aligned}$$

When the response is continuous, it is possible to calculate $\mathbb{E}(Y_U|\boldsymbol{X}, Y_O, \mathbb{G})$ for a set of response values, as shown in earlier calculations. When the response is binary, whether this is possible will depend on the circumstances. Therefore, for the sake of consistency, *Deterministic SP* predicts one value at a time using $\mathbb{E}(Y_i|\boldsymbol{X}, Y_O, \mathbb{G})$ for $i \in U$. When some members of X_U are missing, these calculations can be performed by introducing missing nodes as just discussed.

If these predicted values are being used for cross-validation, the analysis should, in theory, be run once for each training set of samples, potentially returning a different $[\hat{\boldsymbol{I}}]$ each time. Alternatively, the method approximates LOOCV by calculating $\mathbb{E}(Y_i|X_i, X_{-i}, Y_{-i}, [\hat{\boldsymbol{I}}])$ using the same $[\hat{\boldsymbol{I}}]$ for each, obtained by first analysing the data for all samples.

Forced Inclusion

I consider a major advantage of *Sparse Partitioning* its ability to assign individual prior probabilities of association, allowing knowledge from previous experiments to be incorporated. In particular, it is possible to set a predictor's probability of inclusion to 1, indicating that it is certain to contribute. This predictor will necessarily find its way into the current model because all partitions which do not declare it associated will have a prior probability of zero. Equivalently, *Sparse Partitioning* can be supplied with a list of certain associations which likewise are included in every partition / equivalence class considered. This strategy is preferable when multiple copies of each predictor are allowed, as it does not insist a prior probability of 1 for all subsequent copies.

Forced inclusion allows the user to appreciate the presence of known associations which, if not accounted for, might otherwise obscure the detection of novel ones. An alternative strategy is to "regress out" the contribution of known associations in advance, but this assumes their

joint contribution is independent to that of the remaining predictors. Therefore, I believe *Deterministic SP* handles this situation more elegantly. In particular, the user might choose to assign negligible prior probabilities for all unknown associations. In this case, no novel associations will appear in the final equivalence class, but the posterior probabilities will show the relative evidence for each being included. Furthermore, the list of top partitions will report the most likely configuration of the known associations, having exhaustively considered all the ways they might contribute.

Manipulative Search

Suppose a run of *Deterministic SP* concludes that there are two associated predictors, but the user would like a list of the four predictors most likely to be associated. One solution would be to scan the list of posterior probabilities for the two next highest. This will pick predictors according to their marginal probabilities, conditional on the final model, which may not be appropriate. For example, if there is a group of highly correlated predictors with posterior probabilities just shy of 1/2, this approach will pick two predictors from this group, although it is highly unlikely that both are associated. We would prefer to pick just one, then reassess the situation with this predictor included.

Therefore, after completing the standard search, *Deterministic SP* offers the option to forcibly set the number of associations found, similar to how frequentist methods might offer a "prune" feature. The first run of the algorithm will stop when all models in the current neighbourhood score lower than the current one. Consider the effect of increasing all prior probabilities in a uniform fashion. Relatively speaking, the scores of those models which remove a predictor will decrease, the scores of those which add a predictor will increase, while those which swap in a predictor will remain unchanged. Therefore, by gradually increasing the prior probabilities, the highest scoring model out of those which add a predictor will eventually outscore the current model.

The amount by which each probability must be increased can be calculated explicitly by dividing the current model score by that of the highest scoring model with a predictor added. Suppose this threshold equals c. To force the addition of a predictor, it is necessary to replace each p_g with p_g^*, such that

$$\frac{p_g^*}{1-p_g^*} = c\frac{p_g}{1-p_g}.$$

At this point, the algorithm can be restarted from the current state. Immediately, it will add in an extra predictor, then continue using the revised prior probabilities. This tactic can be repeated until the required number of associations are declared. If, instead, it is necessary to remove a predictor, the opposite strategy can be used, each time reducing the

prior probabilities.

5.3 Simulated Data

In this section, I show how *Deterministic SP* performed in a selection of the simulation studies considered in Chapter 4. As the simulated datasets were relatively small (generally, $n = 100$ and $N = 1000$), the run time of *Deterministic SP* was essentially instantaneous, comparable to that of the most basic method, *Single*. In most cases, I compared the method to *Sparse Partitioning*, *Pairs*, *MARS* and "*best other*". For each scenario, *best other* represents the highest score achieved by any of the five remaining methods (*Single*, *CART*, *RF*, *SSS* and *Logic*). The general conclusion is that, while at times *Deterministic SP* suffers compared to the original version of *Sparse Partitioning*, it still managed to match or better the (combined) performance of many available methods.

Figure 5.1 presents results from the first three simulation studies. The plots in the top row, corresponding to Study One (continuous response, idealised data), are very representative of all others. In general, *Sparse Partitioning* and *Deterministic SP* were evenly matched, occupying the top two spots across scenarios. However, on a few occasions, the performance of the deterministic version dropped and, despite remaining above *best other*, the method was been beaten into second place by *Pairs*. The dashed lines in the top plots, indicating the highest pairwise interaction probabilities (conditional on the final model), provide insight into *Deterministic SP*'s results. Its drop in performance under Model III occurred when the method became unable to accurately detect the pairwise interaction.

The figure's second and third rows refer to Study Two (causal predictors unobserved, $r^2 = 0.9$ or $r^2 = 0.8$). As before, average detection was reduced when the causal predictors were not observed directly, however, the ordering of methods remained unchanged. The fourth row corresponds to Study Three (10% of predictors missing) and suggests that *Deterministic SP*'s *ad-hoc* method for obtaining a marginal likelihood in the presence of missing data is effective.

The top two rows of Figure 6.3 relate to Study Four (exponential noise, then uniform noise). Once more, it appears that the nonlinearity of *Deterministic SP*'s underlying relationship model makes the method robust when the assumption of normally distributed residuals is violated. The third row refers to Study Five (tertiary predictors), while the bottom row corresponds to Study Six (binary response). A similar pattern is evident; *Deterministic SP*'s performance has been affected by the restrictive nature of its search, but its consideration of more general underlying relationships still gives it an edge over most existing methods.

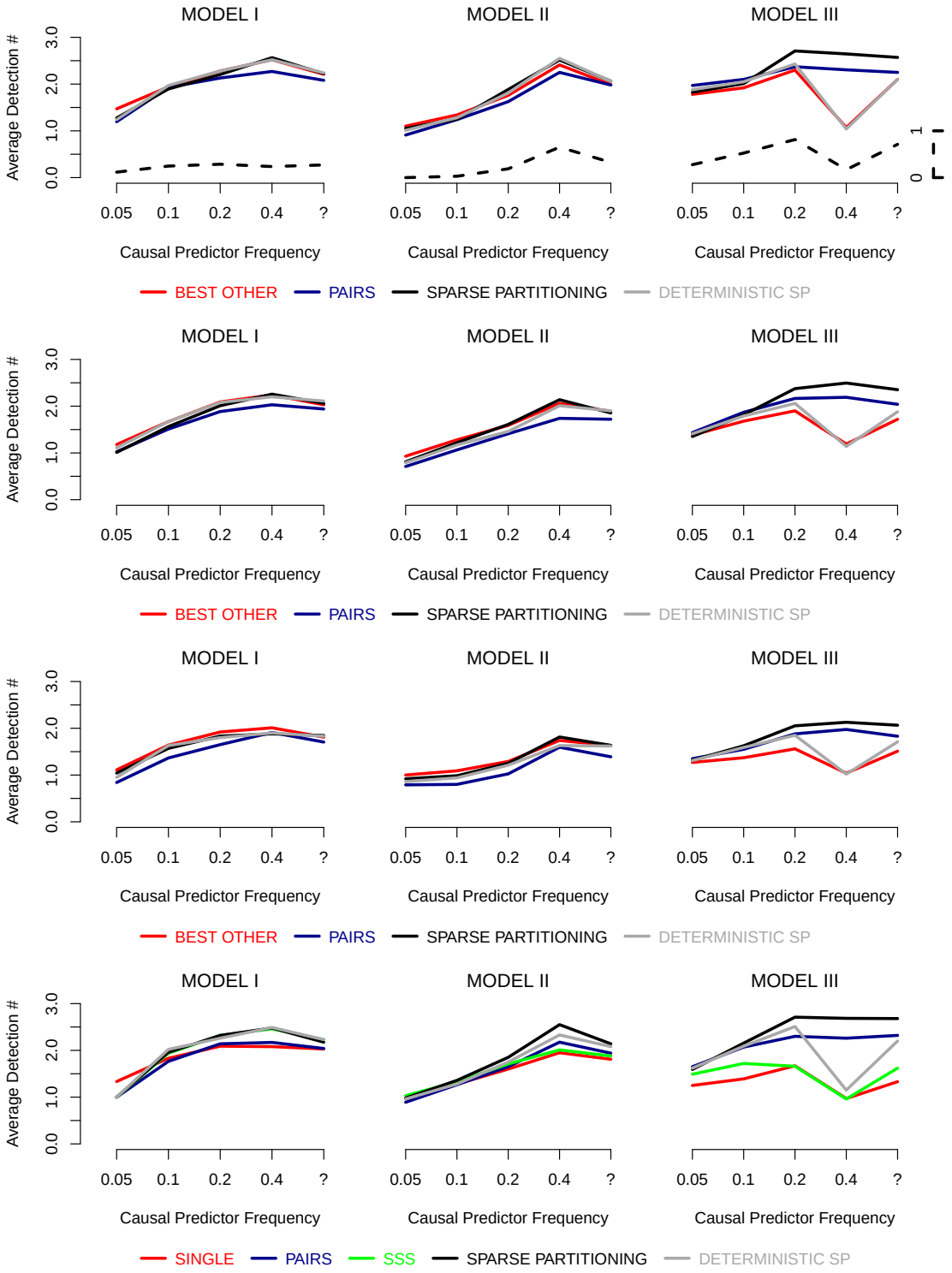


Figure 5.1: *Results of Simulation Studies One to Three. The top row corresponds to Study One (continuous response, idealised data), the middle two rows correspond to Study Two (causal predictors unobserved, $r^2 = 0.9$ or $r^2 = 0.8$), the bottom row corresponds to Study Three (10% missing predictors).*

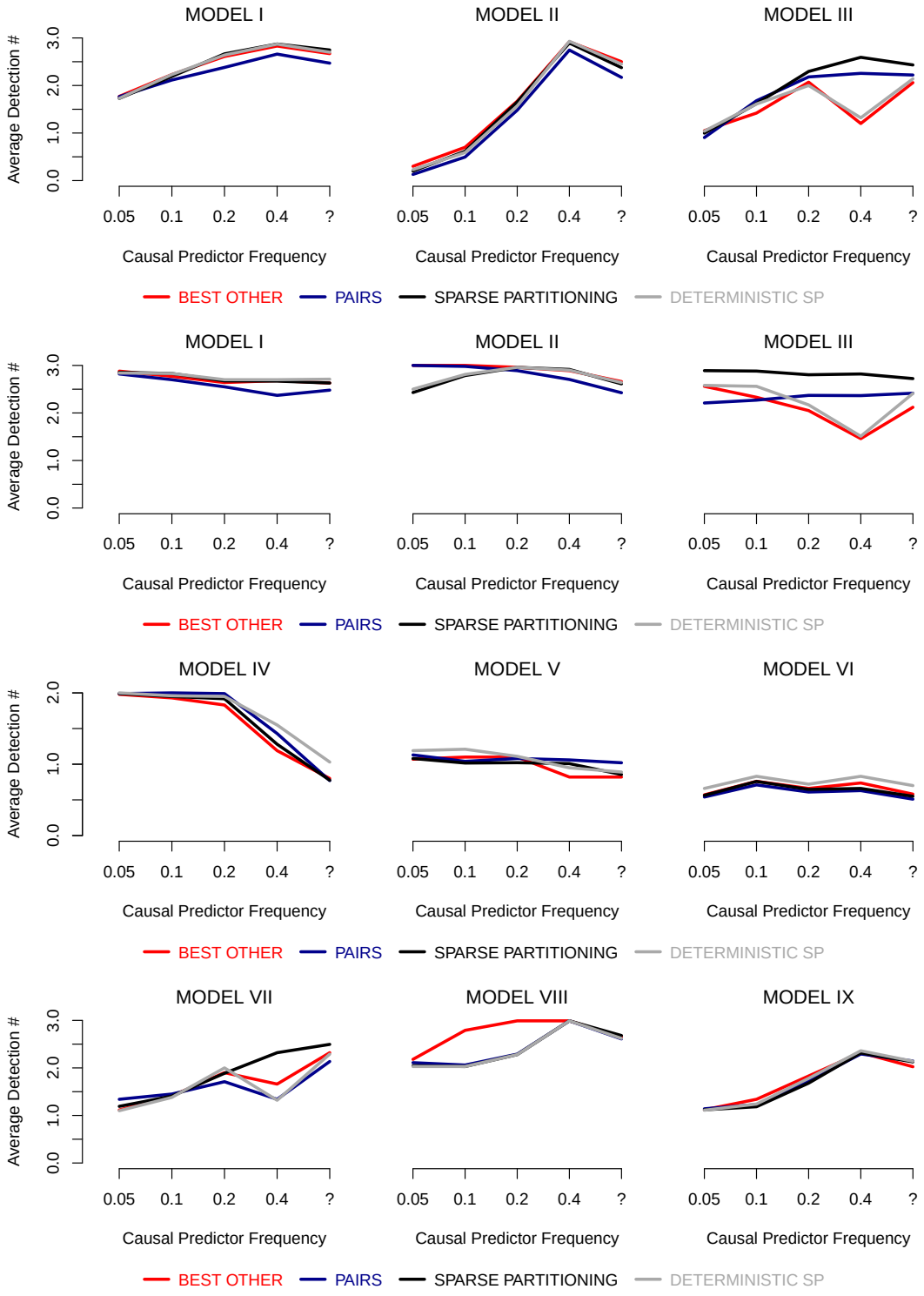


Figure 5.2: *Results of Simulation Studies Four to Six. The top two rows correspond to Study Four (first residual noise was generated from an exponential distribution, then from a uniform distribution), the third row corresponds to Study Five (tertiary predictors, 2 of which were causal), the bottom row corresponds to Study Six (binary response, 3 or 4 causal predictors).*

Like many of its rivals, *Deterministic SP* experienced a dip in performance under Model III for the highest causal predictor frequency (0.4). This suggests that the dip in form is a consequence of using a deterministic search algorithm. When the model space is searched stochastically, the algorithm will have the chance to consider a number of paths leading to the true underlying relationship. Therefore, if the first approach is unsuccessful, a later one might have more luck. For a deterministic search, the algorithm will generally have only one chance to reach the true model, so how favourable this single approach is, will determine how successful the method.

The algorithm of *Pairs* was less, but not completely, immune to this dip in performance, leading it to outperform *Deterministic SP* for a few scenarios. This is disappointing. However, I take consolation from *Pairs*' limitations. Firstly, although it too can be applied to the very largest datasets, extensions to allow, say, three or four way interactions would rapidly become infeasible. Secondly, the underlying relationship is unable to take into account multicollinearity. Just like *Single*, if a group of predictors are strongly correlated with a causality, the top list of pairwise associations will reflect this.

Thirdly, it is very easy to construct situations where *Pairs* will produce misleading results. For example, suppose one predictor has a very strong effect on the underlying relationship; all pairwise models containing this predictor will perform very well, whether or not the second predictor is associated. Fourthly, the method gives little indication whether the top scoring pairs of predictors are contributing additively or via an interaction. *Pairs* could readily be amended to perform tests of "true interaction", by using the additive model as the null hypothesis. The maximum likelihood test would then compare

$$f(X_g, X_{g'}) = \theta_{X_g} + \theta_{X_{g'}} \quad \text{with} \quad f(X_g, X_{g'}) = \theta_{X_g X_{g'}},$$

so the greater the difference between the fit of these two models, the stronger the evidence for an interaction. However, while this change would tackle the last two problems, it would be more difficult to overcome the first two.

5.4 Real Data

In this section, I apply *Deterministic SP* to two whole genome datasets. I look once again at *Arabidopsis thaliana*, this time using data from the latest release of "Project 2010". I then apply the method to data from the METABRIC study, a large-scale collaborative examination of breast cancer.

5.4.1 2010 Project: Release 3.04

In Chapter 4, I examined the 2010 Project's pilot dataset. I now consider data from its most recent release, the subject of a paper by Atwell *et al.* (2010). The expression level of the FLC gene is known to be affected by polymorphisms in the FRIGIDA region (Johanson *et al.*, 2000; Shindo *et al.*, 2005). Atwell *et al.* performed a one-SNP-at-a-time association study using FLC expression as the response. Their analysis produced results similar to *Single*, shown in the top plot of Figure 5.3. While some SNPs within the FRIGIDA region (which is marked by a red vertical line) achieved genome wide significance, two stronger groups of associations were detected approximately 200 kbp and 1 Mbp to the right. Prior knowledge would suggest these downstream associations are spurious. When Atwell *et al.* repeated the analysis, but this time including in the regression model two alleles of the FRIGIDA gene known to affect FLC, the downstream associations vanished, increasing suspicion that they were false positives. In order to draw conclusions, for the remainder of this section I assume this suspicion to be true.

I was interested to see how *Deterministic SP* would cope with a dataset of this size and also how it would fare with the problem Atwell *et al.* observed. To begin this analysis, it was necessary to reconsider the topic of confounding. In May 2010, I was able to visit Vienna and talk over aspects of the paper with members of the Nordborg Lab in person. A striking difference between our approaches was that they had chosen not to correct for population structure. While I viewed any effects attributable to population differences as environmental noise, and so treated these as confounding, they considered that the correlations with geography might have a genetic basis. For example, I had assumed that the Scandinavian accessions had longer flowering times as a result of their cold climate relative to the other accessions. However, it could be that these plants required a different flowering time to survive in these conditions, leading them to adapt genetically to their surroundings. Further doubt was cast over my initial theory when it was pointed out that, for the purpose of the experiment, most plants had been cultured in a climate-controlled laboratory in the basement! Thereafter, I decided it best to correct only for relatedness. Although not shown, my results for the pilot study remain unchanged, as the signal stayed sufficiently strong, regardless of what correction was applied.

In total, 174 samples were recorded for FLC expression levels and each typed for 216,130 SNPs. Of these, I removed 8 accessions on account of extremely high relatedness. For the remainder, I estimated the kinship matrix $\boldsymbol{K}$ using the technique I discussed for the pilot dataset. I then calculated the Cholesky decomposition of $\boldsymbol{K}$ and supplied the columns of this to my method as confounding variables. For each SNP, I set $p_g = 5/216130$, indicating a prior belief of 5 associations. With *Arabidopsis*, for which the sense in the Nordborg Lab was that many tens of predictors might affect a given trait, this should certainly be interpreted as a

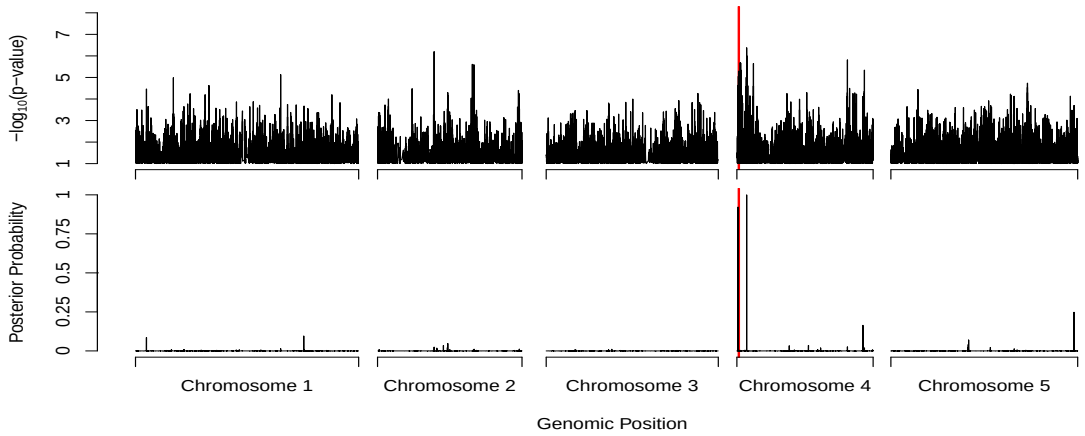


Figure 5.3: *Analysis of* `FLC` *expression. The top plot shows the p-values obtained by* Single*, the bottom plot shows the posterior probabilities calculated by* Deterministic SP*. The vertical red line indicates the location of the* `FLC` *gene.*

prior on the number of strong associations.

The bottom plot of figure 5.3 shows the findings of *Deterministic SP*. The method declared two associations (those with posterior probabilities greater than 0.5). It is clear that the sparsity in the prior carried over into the results, as most posterior probabilities were very close to zero. Figure 5.4 examines the start of Chromosome 4 in closer detail. The problem encountered by Atwell *et al.*, is shown more clearly in the top plot. The nearest peak to the `FRIGIDA` gene was approximately 100 kb upstream. However, the highest p-value in this group was eclipsed by a second and third set, located between 200 kb and 1 Mbp to the right. *Single* also suggested evidence for a fourth set of associations, almost 2 Mbp downstream. The results of *Deterministic SP* appear promising. While it also found strong evidence of an association from the second set, in terms of posterior probabilities, this was followed very closely by one from the first set, that nearest to `FRIGIDA`.

This analysis suggested that not only is the method computationally feasible for datasets of this size, which incidentally took only a few minutes to process, but also that it can produce meaningful results. However, it also alerted me to a possible practical issue when presenting results from *Sparse Partitioning*. In association studies, although linkage disequilibrium can be a nuisance when trying to pinpoint the most likely causal variant, it can also be quite reassuring. When neighbouring predictors are very highly correlated, we expect the results of one-predictor-at-a-time analyses to show broad peaks surrounding any strong association (c.f. the top plot of Figure 5.4). If these patterns are present, this adds an element of replication to the finding. By contrast, if these analyses return only isolated hits, the experimenter might become sceptical, and consider these "finds" were an artifact of genotyping error. It would be

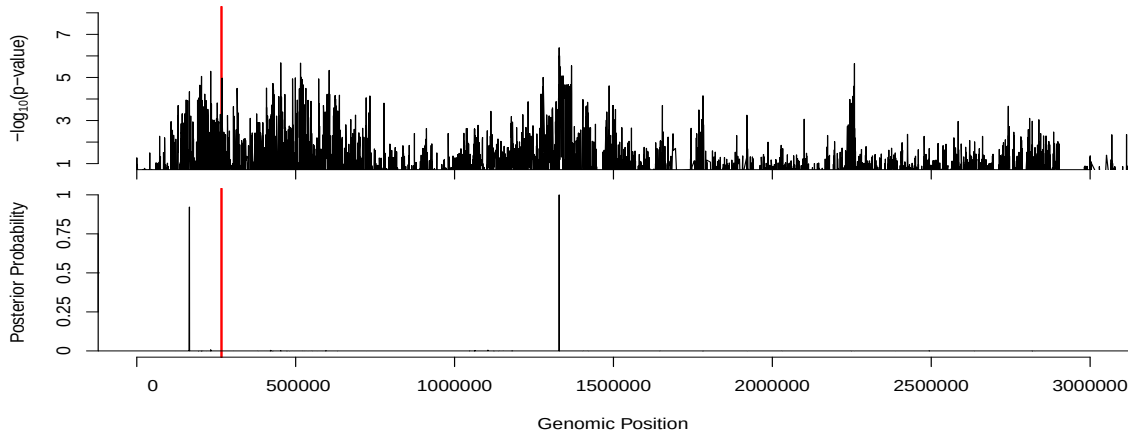


Figure 5.4: *Analysis of* `FLC` *expression. These are the same plots as in the previous figure, except zoomed in on the start of Chromosome 4. The top plot shows the results of* Single*, while the bottom plot reports those of* Deterministic SP*. The red line indicates the location of the* `FRIGIDA` *gene.*

wrong for a user of *Sparse Partitioning* to draw the same conclusions, as the peaks will more likely than not be isolated. However, it seems prudent to always present the results of my method alongside those of *Single*. For example, although the user might be suspicious of the sparse peaks in the bottom plot of Figure 5.4, by referring to the top plot, they will see that these fall within areas which show consistent, marginal association.

5.4.2 METABRIC

I've been fortunate to be involved with METABRIC (Molecular Taxonomy of Breast Cancer International Consortium). This is a collaborative project between the UK and Canada tasked with improving our understanding of breast cancer. In total, over 2,500 individuals have been examined, providing a wealth of data. To give an idea of scale, for the 997 individuals which form "Dataset I", I had access to 906,600 SNP probes, 1,876,300 "Copy Number" probes and measurements of expression levels for 48,803 genes.

Copy Number

Copy number change is a phenomenon which has recently received increased attention (The Wellcome Trust Case Control Consortium, 2010). Consider a reasonably long section of human DNA located on an autosomal (non-sex) chromosome. If this section is unique then, modulo minor mutations (e.g. SNPs), we expect it to appear twice across the genome, once in each copy of the homologous pair to which it belongs. However, this will not necessarily be the case. In some genomes, either one or two sequence copies will have been deleted, while in others, one or more copies will have been added. It is easy to appreciate why copy number variation, a phenomenon disrupting sizeable regions of the genome, might be important.

In METABRIC, these copy number changes are divided into two categories: copy number variants (CNVs), which are "germline" events common to all cells in the body; and copy number aberrations (CNAs), which are "somatic", specific only to tumours. Generally speaking, CNVs are more focal, affecting perhaps a few hundred base pairs, compared to CNAs, which might involve up to a few million.

Preprocessing

Before it was possible for me to analyse the data, I carried out a number of preprocessing steps. The top left grid in Figure 5.5 represents the raw data. Each row relates to an individual, while each column refers to a SNP or copy number probe. Although copy number is integer valued, it is measured on a continuous scale. Black dots indicate probes judged to differ sufficiently from the reference sequence. A segmentation algorithm is applied to determine which probes relate to regions of copy number change. This takes advantage of prior knowledge of the average length of change regions in order to merge values for neighbouring probes. Finally, to determine whether a region corresponds to a CNV or CNA, cancerous cells are compared to healthy ones; the changes which appear in both are likely to be germline, those specific to tumours are likely to be somatic.

My analysis began with the top right grid. Copy number segments had been merged and variants had been typed. It was possible that two or more predictor types coincided at a particular location. Although all data supplied to me was complete, any time a CNV or CNA overlapped with a SNP, I set the latter to missing. This was to recognise the uncertainty involved in calling a SNP in an area disrupted by a copy number change. My first step was to reconstruct the complete probe matrices by separating out predictor types. For the SNPs, their value was already tertiary; 0, 1 and 2 corresponding to homozygous wildtype, heterozygous and homozygous mutant, respectively. For the copy number values, I used 0, 1, and 2 to represent a deleted, neutral or amplified state, relative to the reference genome. At this point, I performed basic filtering, searching first for trivial predictors, then for identical neighbours. This produced the "raw" predictor sets for CNVs, CNAs and SNPs (sizes 11,538, 193,872 and 874,649). Next, I thinned each set using an r^2 threshold of 0.8, which left me with the "processed" predictor sets (sizes 6,328, 11,735 and 523,943).

A study the size of METABRIC offers the potential to ask very detailed questions about breast cancer. However, at the moment, the primary aim is to get a better understanding of the general landscape of the disease. In particular, the study is interested in interrogating the relative contributions of CNVs, CNAs and SNPs, and how they might interact. A primary means for assessing each predictor's regulatory effect is to consider their effect on gene

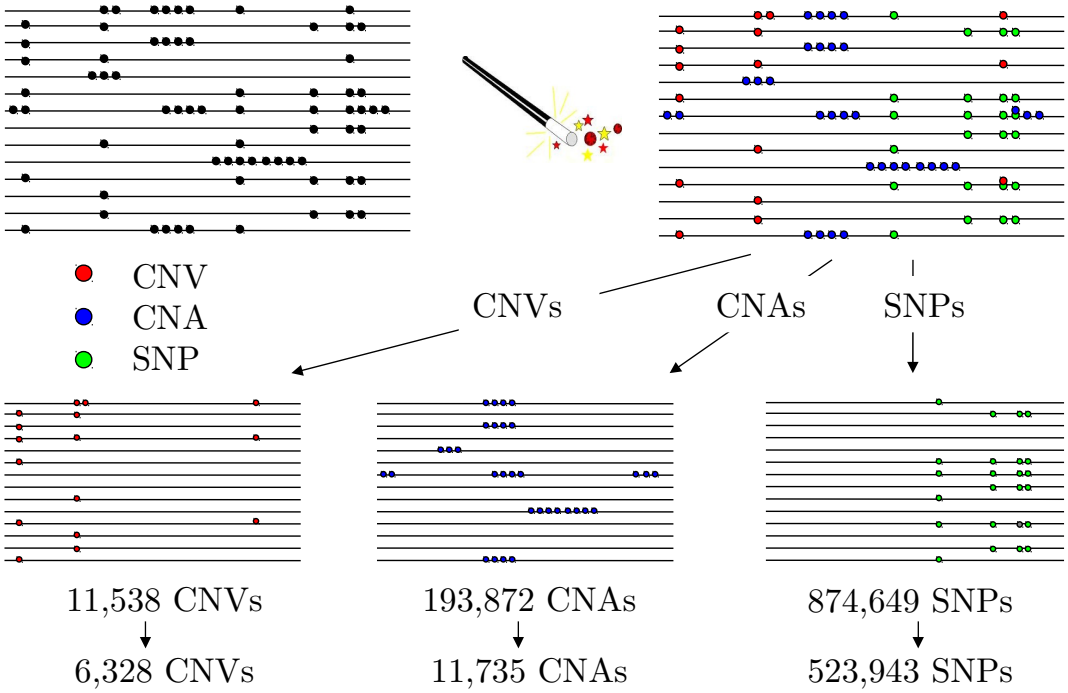


Figure 5.5: *Processing of predictors. The top left matrix visualises the output from the sequencing arrays. Each row corresponds to an individual's genome, while each black dot indicates a locus whose state differs from the reference sequence. The job of the segmentation algorithm (magic wand) was to classify all mutations as CNVs, CNAs or SNPs (red, blue or green dots). At this point, I entered the analysis. It was possible for mutations to be classed as more than one type, indicated by two overlapping coloured dots. However, whenever a copy number change overlapped with a SNP, I set the latter's value to missing, to recognise the difficulty in determining genotypes in this situation. To begin filtering the predictors, I first separated by type, creating individual predictor matrices for CNVs, CNAs and SNPs. Using the CNVs as an example (bottom left), the basic processing step involved searching for trivial predictors, then identical neighbours, and resulted in the "raw" predictor set of size 11,538. Next, I pruned using an r^2 threshold of 0.8, leaving me with the "processed" predictor set, containing 6,328 CNVs. I repeated this process for CNAs and SNPs.*

expression. Translation is the process whereby protein is synthesised according to a gene's DNA sequence. An intermediary of this process is "messenger RNA" (mRNA). Measuring the quantity of mRNA produced by a specific gene provides a reasonable indicator of this gene's activity. METABRIC recorded measurements for 48,803 gene probes in total. After basic quality control, for example, removing probes with low confidence scores and those matching more than one genomic location, I was provided with a total of 28,609 sets of measurements.

The samples were obtained from three different sites: Cambridge, Nottingham and Vancouver. To investigate the effect of site as a confounder on expression values, I applied principal component analysis to the full set of responses. Borrowing notation from the multiple response case (presented in Chapter 3), each column of $\boldsymbol{Y}$, which is now a matrix (size $997 \times 28,609$), corresponds to a particular gene probe. Principal component analysis of $\boldsymbol{Y}\boldsymbol{Y}^T$ calculates the linear combinations of responses across which individuals show most variation. If, for example, all individuals measured in Cambridge had on average higher expression values, we would expect this to be evident in the top principal component axes. The left plot of Figure 5.6 demonstrates the noticeable impact site had on expression values, with the second principal axis being a particularly good indicator of clinical centre. Similarly, a factor which has a well-known impact on gene expression values is "ER status", a measure of an individual's levels of estrogen. The right plot presents the same principal component values, but this time individuals are coloured according to whether they are ER positive or ER negative. It is evident how well the first principal axis distinguishes these two groups. These observations led me to regress out the contribution of site and ER status in advance of analysis. Essentially, I created a 3 by 2 contingency table and, for each response, subtracted the mean expression value for each cell from each of the corresponding individuals.

Basic Analysis

My first analysis, applying *Single* to each predictor-response pair, was incredibly naïve, but provided a good feel for the nature and size of the data. For each expression, I recorded the smallest p-value for each set of predictors. These tests immediately highlighted the extent of, and threat posed by, outliers. Considering many of the predictors had either one or two rare states, the presence of extreme expression values heavily exaggerated p-values. Stranger *et al.* (2007) discovered a similar problem in the dataset they studied, and chose to combat the issue via permutation testing. However, I decided on our scale this would be impractical if decent resolution was desired in a reasonable time. I considered attempting to remove or moderate extreme response values, but such an approach seemed too subjective. Therefore, I decided to replace all expressions with their ranked values, so that each response became a permutation of $\{1, 2, ..., 997\}$.

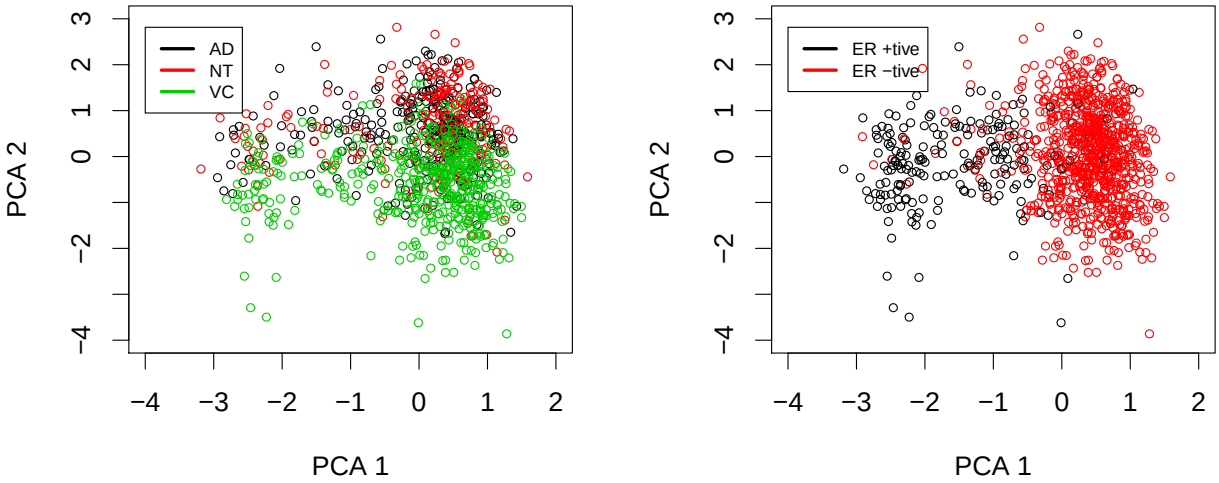


Figure 5.6: *Influence of confounders. In each plot, the x and y axes correspond to the first and second principal components through the response values. In the left plot, samples are coloured according to site of origin (AD = Addenbrookes Hospital, Cambridge; NT = Nottingham; VC = Vancouver). In the right plot, samples are coloured according to ER status. The impact of both site and ER status is clearly visible through these axes.*

To account for the multiple testing involved in comparing all predictor-response pairs, I applied a Šidák-type correction (Šidák, 1967) to the minimum p-values. I based this correction on N_e, an estimate of the effective number of tests performed (specific to each predictor type), correcting each p-value with the transformation: $p \to 1-(1-p)^{N_e} \approx N_e p$, for small p. Similar to the technique I used when analysing *Arabidopsis* data (Figure 4.11), I determined suitable values of N_e by studying quantile-quantile plots. Using CNVs as an example, I began with the set of 28,609 minimum p-values, each one obtained by regressing a particular expression on each of the 11,538 raw CNV predictors in turn. I then plotted, for different possible values of N_e, the set of transformed p-values, choosing the value of N_e which resulted in a line closest to the diagonal. In the event, this suggested the effective numbers of tests were approximately 3,200, 9,200 and 701,000, for CNVs, CNAs and SNPs, respectively. Reassuringly these values were fairly similar to those obtained by filtering using an r^2 value of 0.8.

The left diagram of Figure 5.7 provides an overview of *Single*'s results. In total 11,162 of the genes tested (39.0%) were found to be associated with either a CNV, CNA or SNP at a 0.0001 significance threshold. This Venn diagram divides these genes according to the types of predictors declared associated. For example, the top left circle corresponds to CNVs. This indicates that 80 genes in total were found associated with CNVs, of which 36 also had associations with CNAs, 13 with SNPs and 28 with both. The most dominant values relate to CNAs, demonstrating the dramatic impact these variants have on gene expression within tumour cells. The tests showed that CNAs were involved in 97.0% of genes found to be associated with at least one predictor type, compared to 0.72% and 14.2% for CNVs and

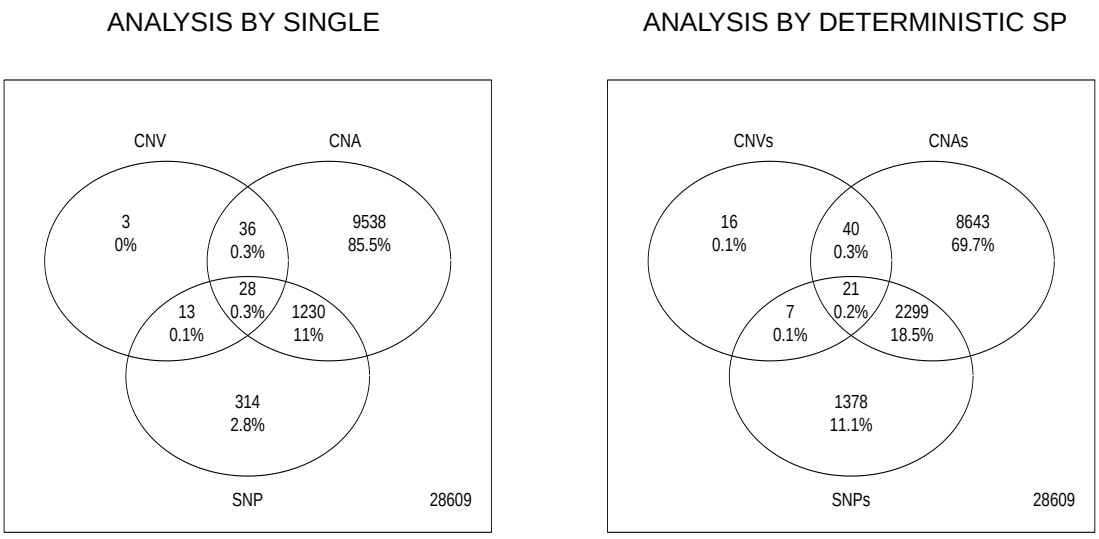


Figure 5.7: *Comparison of the results of* Single *and* Deterministic SP*. The left Venn diagram relates to predictor-gene pairs found associated by* Single*, the right relates to* Sparse Partitioning*. The values report the number of times an association was found between a gene and one or more CNVs, CNAs and/or SNPs. Each section in the Venn diagram corresponds to a particular combination. For example, the top left circle of the left diagram indicates that 80 gene expressions were found to have an association with a CNV. Of these, 36, 13 and 28 also had an association with a CNA, with a SNP or with both a CNA and SNP.*

SNPs, respectively. This domination is perhaps to be expected, considering that CNAs were defined as variants unique to cancerous cells.

Analysis by Deterministic SP

It is potentially of great interest to identify interactions between predictors, as doing so might suggest common pathways. In particular, if a provable interaction between a germline and somatic variant was found, this would suggest an individual's inherited background influences subsequent alterations during cancer. For this reason, there is a demand for tools able to consider large numbers of predictors whilst investigating interactions (Gilad *et al.*, 2008).

My strategy was to apply *Deterministic SP* for each gene, regressing its values simultaneously on CNVs, CNAs and SNPs, using the sets of processed predictors. I might have used the sets of raw predictors instead, however, filtering reduced a considerable amount of work. It also helped in specification of prior probabilities of association. The pruning of predictors according to r^2, to some extent, standardised the predictor variation, which could be considered appropriate when assigning p_g based on an expected average number of associations. For example, if for CNAs I had expected 5 associations and set $p_g = 5/193872$, this could be viewed as overly strict, when the effective number of CNA predictors was closer to 11,735.

There were 542,006 processed predictors in total. However, being a preliminary analysis, I decided to make this number more manageable still. Therefore, for each expression, I considered only those SNPs on the same chromosome as the relevant gene, while including all CNVs and CNAs. This left me with between approximately 30,000 and 60,000 predictors for each experiment. Hopefully, this did not take too much away from the analysis, as the majority of associations found so far have been within *cis* (Stranger *et al.*, 2007).

I applied *Deterministic SP* to each experiment twice. I set the prior probabilities of association to $\frac{1}{3}/6328$, $\frac{1}{3}/11735$ and $\frac{1}{3}/523843$ for CNV, CNA and SNP predictors. This was designed so that there would be on average $\frac{1}{3} + \frac{1}{3} + \frac{1}{3} = 1$ association per gene. However, because I only used a subset of SNPs for each expression, the actual expected number of associations would be slightly lower. I felt it was better to keep these values, rather than increase the prior probabilities for SNPs, for the reason that I had retained the *cis* predictors, those most likely to be associated.

For both runs, I allowed up to 8 causal predictors. Post-analysis checking suggested this was a reasonable limit, as on only two occasions was it reached. I restricted the first run to the linear underlying relationship, $S = 1$, setting the maximum number of groups K to 8. For the second run, I allowed for two-way interactions, setting $S = 2$ and $K = 4$. In total, each run took approximately 1,500 hours, which across a multi-core cluster meant it was, by-and-large, achieved overnight. There was a considerable difference between the run times for different expressions. Those with 0, 1 or 2 associations would be completed in less than a minute, but those with 5 or more took up to a few hours.

Similar to the analysis by *Single*, it was possible to divide the genes according to with which types of predictors they were found associated. The right diagram of Figure 5.7 displays the corresponding Venn diagram for the linear version. In total, the linear run of *Deterministic SP* found 12,404 genes (43.4%) to have one or more associations, which was reasonably similar to the figure of 11,162 (39.0%) produced by *Single*. This is to be expected, as the first step of *Deterministic SP* looks for marginal associations. However, it is interesting to note how the composition of associations changed. CNAs continued to dominate the expression landscape, as they did in the one-predictor-at-a-time analysis, being involved for 88.7% of genes with at least one association. For SNPs this figure was 29.9% (up from 14.2%), while for CNVS it remained roughly the same at 0.68% (compared to 0.72%). These figures suggest that allowing for multiple types of predictors in the model assisted the detection of SNPs, by first accounting for the strong effect of CNAs. Although less striking, this effect was noticed for CNVs as well.

The following table compares the linear and nonlinear runs of *Deterministic SP* by how many associations were found for each gene:

	Number of Predictors Declared Associated								
Settings	0	1	2	3	4	5	6	7	8
$S = 1; K = 8$	16205	5669	4123	1836	600	136	31	7	2
$S = 2; K = 4$	16205	5749	3991	1871	599	159	27	8	0

The total numbers of genes found to have an association by each run were necessarily equal as, regardless of the values for S and K, the first step of the algorithm is to see whether a single predictor crosses the posterior probability threshold. The main difference between the two runs is that the nonlinear version declared two predictors associated on 132 fewer occasions. However, it was fairly even whether the corresponding 132 genes ended up with only one or more than two associations, and, as a result, the total numbers of associations found by each run were almost equal (22,754 compared to 22,753).

To gauge evidence for interactions, I merged the posterior probabilities for interactions across all experiments for the nonlinear run. For the genes for which two associations were found, *Deterministic SP* will have returned 1 pairwise interaction probability; for those with three associations, it will have returned 3; for those with four associations, it will have returned 6, and so on. In total, this produced 15,361 (non-distinct) pairwise probabilities.

Of these probabilities, some were very high; 217 pairs exceeded $1 - 10^{-5}$, while 3 exceeded $1 - 10^{-10}$. From these values, can we automatically conclude that there is strong evidence that expression levels are influenced by interactions between predictors? As I discussed earlier, there is an inherent difficulty in trying to assign significance to interaction probabilities. The most robust assessment would be permutation testing, but there is no obvious way to perform this, even if computational limitations were not an issue. On a positive note, it helps that there is a degree of independence between the posterior marginal probability estimates and the posterior interaction probability estimates. Each set of pairwise probabilities was calculated conditional on the final equivalence class. It compared those partitions which featured the interaction to those without, and so did not explicitly depend on the final equivalence class' score. This meant that a high scoring equivalence class would not necessarily return strong pairwise interaction probabilities, and vice versa, suggesting the interaction probabilities can be assessed on their own merits.

Fortunately, in this case, we are able to interrogate the values further. We strongly expect interactions within the genome to act locally. However, at the moment, it is not possible to accommodate such a belief within *Sparse Partitioning* directly. Therefore, if *Deterministic SP* found strong support for such a pattern, this would offer evidence for the interaction probabilities being valid. Again, there are caveats. Firstly, I only considered SNPs on the same

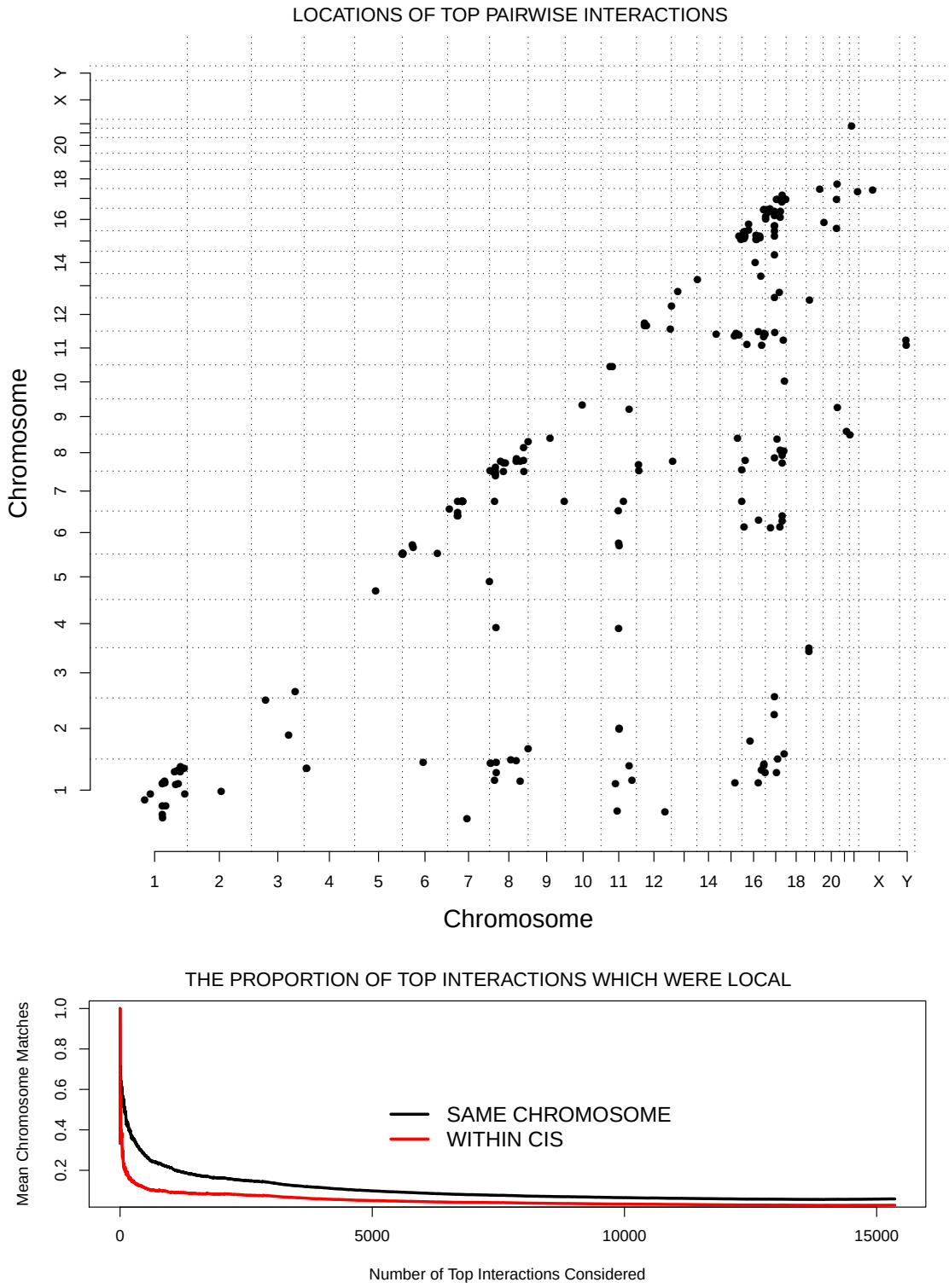


Figure 5.8: *Pairwise interactions, collated across all genes. The top plot shows the location of the top 217 pairwise interactions (posterior probabilities greater than* $1 - 10^{-5}$*). Both axes indicate genomic position. The bottom plot displays the mean number of "local" pairwise interactions, when considering different subsets of the 15,361 interactions reported by* Deterministic SP*. The black line defines local as on the same chromosome, while the red line defines it as on the same chromosome and within 10 Mbp. To calculate these values, I considered the partial means as pairs of predictors were added from highest to lowest ranking. The local interactions are clearly over-represented within the highest scoring pairs, apparent by the initial elevation of the running mean above its base (far right) value.*

chromosome, so any interaction between a pair of these would necessarily be relatively local. Secondly, it might be the case that *Deterministic SP* was biased towards interactive models in general. Suppose *Deterministic SP* automatically tried to place predictors in as few groups as possible, which is understandable considering these models will always fit better than their linear counterparts. As many of the strong associations will be in *cis*, this would lead to the method returning high probabilities for *cis* interactions, but without basis.

To get around these issues, instead of focusing on absolute interaction probabilities, we can instead consider the spread of all 15,361 values. The top plot in Figure 5.8 shows the pairwise locations of the 217 pairs of predictors with posterior interaction probabilities greater than $1 - 10^{-5}$. It is possible to make out a bias towards points lying on the diagonal. The second plot is more compelling. The black line plots the running mean of the number of pairs lying on the same chromosome, calculated as pairs of predictors are added in decreasing order of posterior interaction probability. For example, when the strongest 50 (100) pairs are considered, 54% (43%) lie on the same chromosome. This is compared to an overall average of 9.6%. The red line plots the same statistic, but this time local is defined as within 10 Mbp. Of the top 50 (100) pairs, 24% (18%) lie within this definition of *cis*, compared to an overall average of 3.5%. These results strongly support the credibility of the posterior interaction probability estimates, and thus lend support for the existence of interactions.

Finally, I highlight one example, that corresponding to the highest pairwise interaction probability reported. This occurred for the gene `VAV3`. Figure 5.9 marks the p-values from *Single* for both the whole genome (top) and a 3 Mbp window surrounding the gene probe (bottom). The y-axis has been spliced at the top, to accommodate the very extreme values detected. In total, *Deterministic SP* declared three associations (black dots), all located within this window, which I have labelled SNP 1, SNP 2 and CNA 1. The strongest association, SNP 1, was positioned very close to the gene probe and received a very high p-value (10^{-164}). Interestingly, this was not the smallest p-value reported, as one SNP's was 69 orders of magnitude stronger. The second SNP also resided near the gene probe. Note that part of the filtering of genes was to remove those which had any predictors located completely within their sequence probe. The third predictor, a CNA, received a relatively weak p-value ($10^{-1.3}$). Therefore, this CNA could only have been declared associated by considering a joint model, one able to examine groups of predictors simultaneously.

Figure 5.10 looks at models involving subsets of these three predictors. The plots in the top row consider the three marginal models for SNP 1, SNP 2 and CNV 1. The bottom left plot considers the pairwise interaction of SNPs 1 and 2, which the bottom right plot examines in more detail. In each boxplot, the final column corresponds to the partition's missing node. Reassuringly, in all boxplots, there is no obvious skew of expression values for the missing

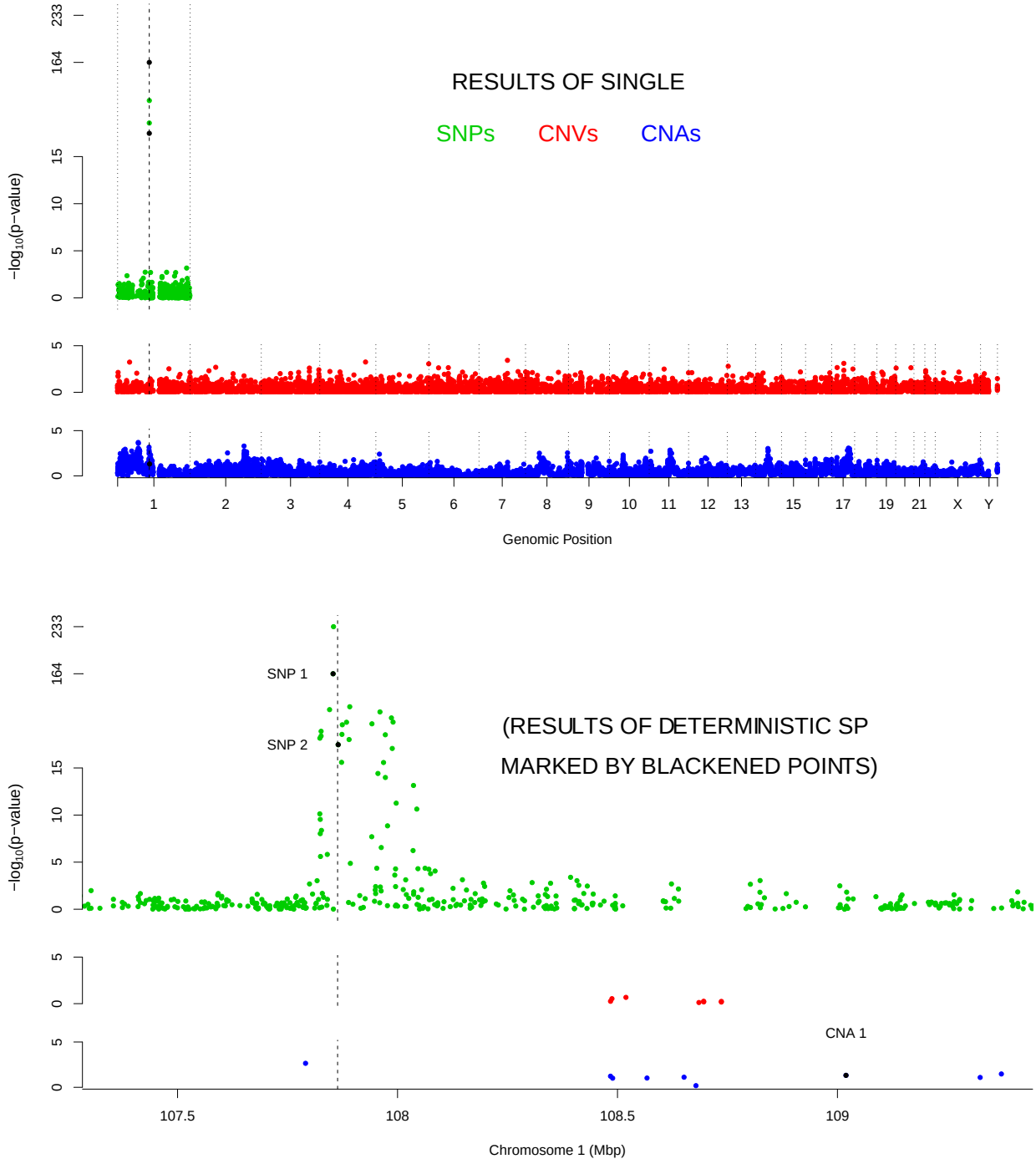


Figure 5.9: *Results for* `VAV3`*. The top plot reports genome wide p-values from the analysis of* `VAV3` *by* Single*. The green points correspond to SNPs, the red to CNVs and the blue to CNAs. Note that, because* `VAV3` *resides on Chromosome 1, only SNPs on this chromosome were considered. The bottom plot presents the same information, except zoomed in on a 3 MBp window surrounding the gene probe (marked by a vertical dashed line). In both plots, the black dots indicate the three predictors* Deterministic SP *declared significant, labelled SNP 1, SNP 2 and CNA 1.*

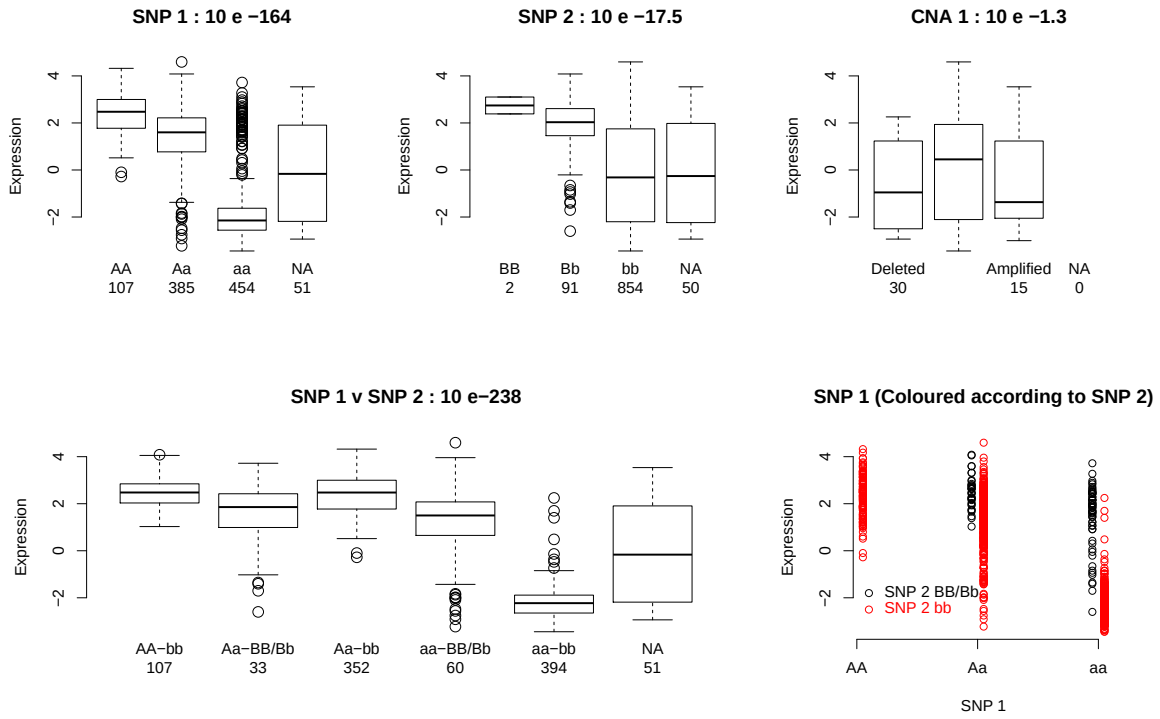


Figure 5.10: *Boxplots for* `VAV3`*. The four boxplots compare the expression levels of* `VAV3` *for different predictor states. The top row displays the marginal effects of SNP 1, SNP 2 and CNA 1, the three predictors declared associated by* Deterministic SP*. The bottom left boxplot considers the interaction of SNPs 1 and 2. In each boxplot, the final predictor state ('NA') corresponds to those individuals assigned to the missing node. The bottom right diagram plots each individual's expression value. The x-axis indicates the value of SNP 1, while the colour of each point indicates the value of SNP 2. It is clear that, although SNP 1 serves as a good predictor of expression value, it is far from perfect, as there remain a number of homozygous mutant (State 'aa') individuals with high expression values. But when we refine these individuals according to which are also homozygous mutant at SNP 2 (State 'bb'), most of the high values are filtered out, suggesting a multiplicative interaction. Note that, as all models shown involve at most two predictors, they only represent approximations of the partitions contained within the final equivalence class, as in that class all three predictors were declared associated at once.*

node, which if present would suggest experimental bias. The strong marginal effects of SNPs 1 and 2 are clearly visible. Note that SNP 2 is recorded homozygous wildtype only twice, so I subsequently merge this state with heterozygous for the interaction boxplot.

The pairwise plots clearly indicate a possible interaction and suggest an interpretation. When both SNPs on their own are homozygous mutant, this has a devastating effect on expression levels. However, for SNP 1 the boxplot picks up a number of "outliers" within the homozygous mutant class, suggesting the marginal model is not ideal. These all but vanish when SNP 2 is taken into account, as clearly evident in the bottom right plot. As the remaining three classes "flatten out", this suggests a multiplicative interaction, triggered only when individuals are homozygous mutant for both SNPs.

Chapter 6

Bayesian Projection Pursuit

Sparse Partitioning *requires that all predictors are tertiary. One way to apply the method to quantitative predictors is by first coercing the dataset into the required format. However, such a strategy is unlikely to be the most efficient as it will not make full use of the information provided by data. In this chapter, I discuss how incorporating aspects from the technique projection pursuit leads to a method able to handle quantitative predictors directly.*

6.1 Motivation

When considering the interaction of groups of predictors, *Sparse Partitioning* explores functions with full degrees of freedom, one for each observed node. This prompts the requirement that all predictors are binary or tertiary, to ensure the number of nodes, and therefore degrees of freedom, remains manageable. When the predictors are quantitative, there are likely to be far more than three distinct values for each predictor. As a result, the number of nodes observed for each group will grow very rapidly, bounded only by the total number of samples. Therefore, any full interaction model is likely to substantially overfit the data, leading to meaningless results.

Furthermore, for each function, *Sparse Partitioning* assigns independent priors to its regression coefficients. This is suitable for categorical predictors, as it reflects that these imply no ordering nor measure, describing only whether two samples' predictor values are the same or different. Quantitative predictors, by contrast, offer far more information. If a quantitative predictor's values are discrete, they will dictate an ordering to the samples. If they are continuous, they will additionally provide a relative measure. In both cases, it is reasonable to expect that when two samples' values at a causal predictor are closer together, their responses will be more similar. This assumption should be reflected in the prior for the functions.

6.2 The Projection Pursuit Model

Once again, we must consider how best to represent and explore interactions. Consider the interaction of two predictors, in which case $f_k(X_g, X_{g'})$ is a bivariate function with a two-dimensional domain. When we examined categorical predictors, the locations of the nodes were irrelevant. This meant that the two-dimensional nature of the function was redundant and allowed us to view the function simply as a mapping of independent points. Now that the predictors are quantitative, the node locations become informative and should be taken into account. Furthermore, while only a finite number of distinct predictor values are observed, these typically represent a small proportion of those possible. Therefore, it makes more sense to consider a *properly* multivariate function, one defined over a wider domain than just the observed nodes.

In order to explore *properly* multivariate functions, it is necessary to first devise a suitable parametrisation, ideally one which retains the generality we desire. The technique projection pursuit (Friedman and Tukey, 1974) suggests a possible solution. The following description comes from Elements of Statistical Learning (Hastie *et al.*, 2001) (ESL), with the notation adapted to match mine:

> The projection pursuit regression equation has the form
>
> $$f(\boldsymbol{X}) = \sum_{k}^{K} f_k(\boldsymbol{X}\boldsymbol{\xi}_k).$$
>
> This is an additive model, but in the derived features $\boldsymbol{X}\boldsymbol{\xi}_k$ rather than the predictors themselves. The functions f_k are unspecified and are estimated along with the directions $\boldsymbol{\xi}_k = (\xi_{k1}, \xi_{k2}, \ldots, \xi_{kN})$ using some flexible smoothing method. The function $f_k(\boldsymbol{X}\boldsymbol{\xi}_k)$ is called a "ridge function" in $\mathbb{R}^N$. It varies only in the direction defined by the vector $\boldsymbol{\xi}_k$. The scalar variable $\boldsymbol{X}\boldsymbol{\xi}_k$ is the projection of $\boldsymbol{X}$ onto the vector $\boldsymbol{\xi}_k$, and we seek $\boldsymbol{\xi}_k$ so that the model fits well, hence the name "projection pursuit".

By considering sums of functions, each acting on a projection, projection pursuit is able to explore a broad range of underlying relationships. In fact, with sufficiently large K, it is possible to approximate, arbitrarily well, any continuous function as the sum of ridge functions. Notice how the model deconstructs the underlying relationship. Potentially, this relationship is a nonlinear, multivariate function of the predictors; using the projection pursuit model, it is expressed as the sum of nonlinear, univariate functions (the ridge functions) acting on linear, multivariate functions (the projections).

ESL suggests an iterative algorithm for evaluating the projection-function pairs within a frequentist framework. Suppose we have picked an objective function and decided what functions are permissible. The algorithm begins by evaluating f_1 and $\boldsymbol{\xi}_1$ recursively. Given the direction $\boldsymbol{\xi}_1$, and thus a projection $\boldsymbol{X}\boldsymbol{\xi}_1$, we wish to calculate f_1 which optimises the objective function. With judicious choice of objective function, this will be explicit. Similarly, given a function f_1, we wish to find the optimal $\boldsymbol{\xi}_1$. Assuming the function is differentiable, Taylor's theorem will provide an approximation of $f_1(\boldsymbol{X}\boldsymbol{\xi}_1)$, linear in $\boldsymbol{\xi}_1$, using which the direction can be updated. These two steps are iterated until satisfactory convergence. Having determined values for $\boldsymbol{\xi}_1$ and f_1, a new projection can be added in and this process repeated.

Popularity of Projection Pursuit

Projection pursuit is not alone in its desire to find informative directions through the data. Perhaps the most popular proponent of this is principal component analysis, although this method seeks to explain variance within the predictors rather than variance within the responses. Nor is projection pursuit unique in its use of smoothed functions (Wahba, 1978). In this respect, it has much in common with functional data analysis, a collection of methods designed to fit curves through data points (Ramsay and Silverman, 2006). Here, the measurements are typically realisations at specific time intervals, so the aim is to find functions which explain the chronological development of an outcome. For these, smoothing functions prove very convenient, as their penalty functions reward desirable properties.

Despite its relative old age, projection pursuit remains a little used technique. This might be due to its model's lack of interpretability. Although it returns an explicit form for $f(\boldsymbol{X})$, from the tangle of overlapping projections and functions it will be difficult to obtain much insight into the underlying relationship. Saying this, projection pursuit can claim to have inspired a number of related methods. For example, neural network methods often consider nonlinear functions of projections, although typically insisting the functions take on a far simpler form (Hastie *et al.*, 2001).

To implement the algorithm described in ESL requires harmonious selection of an objective function and a class of functions. For reasons which become clear, natural cubic splines prove a convenient choice of functional class.

Natural Cubic Splines

The function $f_k(t)$ is termed a piecewise polynomial if it is constructed as a series of polynomials, each defined on contiguous intervals. The values of t dividing these intervals are termed knots. If each polynomial section is limited to order at most three and constructed so that $f_k(t)$ is continuous, with continuous first and second derivatives, the function becomes

a "cubic spline". Often such functions, while well-defined on the interior intervals, become erratic outside of the two extreme knots. To counter this, it is common to insist they are linear on these regions, in which case they become "natural cubic splines". Supposing there are d_k knots, it is straightforward to calculate the degrees of freedom of a natural cubic spline. $4(d_k - 1)$ parameters are required to describe the $d_k - 1$ internal cubic polynomials and 4 parameters to describe the two linear sections. The continuity conditions place three constraints at each knot, and so the total degrees of freedom equals $4(d_k - 1) + 4 - 3d_k = d_k$.

An important principle of *Sparse Partitioning* is that the functions maintain generality. When we considered categorical predictors, this insisted that every node could be mapped to any real value. In the same way, when the predictors are quantitative, we desire that for every knot (distinct projection value $X_i\boldsymbol{\xi}_k$), there is no restriction on the range of values to which it can be mapped. As I will show shortly, if the functions are natural cubic splines, this condition can be fulfilled.

In choosing an objective function, it is desirable to incorporate a measure of smoothness. Given two functions which fit the data equally well, we will prefer the one which is "more smooth". A commonly used smoothness penalty is based on the integral of the second derivative:

$$\text{Pen}(f_k, \lambda) = \lambda \int (f_k''(t))^2 \, \mathrm{d}t,$$

where $\lambda \in [0, \infty)$ is termed the smoothing parameter. Although my version of Projection Pursuit will be applicable to binary responses, it is easiest to explain the underlying concepts for a continuous response. Therefore, consider using the penalised residual sum of squares

$$\text{PSS}(f_k, \lambda, \boldsymbol{Y}) = \sum_i (Y_i - f_k(X_i\boldsymbol{\xi}_k))^2 + \text{Pen}(f_k, \lambda).$$

This leads to the reason for choosing natural cubic splines. Suppose that, for a given λ and realisation of $\boldsymbol{X}\boldsymbol{\xi}_k$, we wish to minimise $\text{PSS}(f_k, \lambda, \boldsymbol{Y})$ across all functions f_k with continuous first and second derivatives. As will be demonstrated shortly, it turns out that this is achieved by a natural cubic spline, with knots defined at each unique value of $\boldsymbol{X}\boldsymbol{\xi}_k$ (Reinsch, 1967).

The following explanation, copied from Pollock (1999), but with notation altered to match mine, explains the construction of a natural cubic spline. This description refers to the kth function, but for convenience I omit this subscript when referring to spline coefficients. For categorical predictors, we were interested in the mapping of distinct values of $X_{i\boldsymbol{G}_k}$ (nodes). In a similar fashion, we are now interested in the mapping of distinct values of $X_i\boldsymbol{\xi}_k$ (knots), except this time, being scalar values, there have a natural ordering. Suppose the set of projected values $\{X_1\boldsymbol{\xi}_k, X_2\boldsymbol{\xi}_k, \ldots, X_n\boldsymbol{\xi}_k\}$ has d_k knots (when the predictors are continuous,

often the number of knots d_k will equal the number of samples n). Let $t_1, t_2, \ldots, t_{d_k}$ represent the ordered set of knots. For $t \in [t_1, t_{d_k}]$, the natural cubic spline f_k can be written as

$$f_k(t) = \theta_d + \gamma_d(t - t_d) + \beta_d(t - t_d)^2 + \alpha_d(t - t_d)^3,$$

where $t_d \leq t \leq t_{d+1}$. The three continuity conditions produce the following constraints, for $d = 1, 2, \ldots, d_k - 1$:

$$\begin{array}{rcrcrcrcl} \theta_d & + & \gamma_d h_d & + & \beta_d h_d^2 & + & \alpha_d h_d^3 & = & \theta_{d+1}, \\ & & \gamma_d & + & 2\beta_d h_d & + & 3\alpha_d h_d^2 & = & \gamma_{d+1}, \\ & & & & 2\beta_d & + & 6\alpha_d h_d & = & 2\beta_{b+1}, \end{array}$$

where $h_d = t_{d+1} - t_d$. Adding in the condition of linearity outside of the boundary knots, which implies $\beta_1 = \beta_{d_k} = \alpha_{d_k} = 0$, the system of equations for the kth function can be written as $\boldsymbol{R}_k \boldsymbol{\beta}_k = \boldsymbol{Q}_k \boldsymbol{\theta}_k$:

$$\begin{bmatrix} r_1 & h_2 & 0 & \cdots & 0 & 0 \\ h_2 & r_2 & h_3 & \cdots & 0 & 0 \\ 0 & h_3 & r_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & r_{d_k-3} & h_{d_k-2} \\ 0 & 0 & 0 & \cdots & h_{d_k-2} & r_{d_k-2} \end{bmatrix} \begin{bmatrix} \beta_2 \\ \beta_3 \\ \beta_4 \\ \vdots \\ \beta_{d_k-2} \\ \beta_{d_k-1} \end{bmatrix} = \begin{bmatrix} q_1 & s_1 & q_2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & q_2 & s_2 & q_3 & \cdots & 0 & 0 & 0 \\ 0 & 0 & q_3 & s_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & s_{d_k-3} & q_{d_k-2} & 0 \\ 0 & 0 & 0 & 0 & \cdots & q_{d_k-2} & s_{d_k-2} & q_{d_k-1} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \\ \theta_{d_k-2} \\ \theta_{d_k-1} \\ \theta_{d_k} \end{bmatrix},$$

where $r_d = 2(h_d + h_{d+1})$, $q_d = 1/h_d$ and $s_d = -(q_d + q_{d+1})$.

As with categorical predictors, the components in $\boldsymbol{\theta}_k$ represent the realisations of f_k at the knots values. Given these, all other coefficients can be calculated. This demonstrates the generality of a natural cubic spline; given any distinct $\{t_1, t_2, \ldots, t_{d_k}\}$ and images $\{\theta_1, \theta_2, \ldots, \theta_{d_k}\}$, a natural cubic spline can be constructed to pass through the coordinates $\{t_d, \theta_d\}$. With this being the case, given a direction $\boldsymbol{\xi}_k$, and hence the projection $\boldsymbol{X}\boldsymbol{\xi}_k$, each spline can be uniquely described by $\boldsymbol{\theta}_k$, the values it assigns each knot, so there is no need to specify the other coefficients.

Continuing through the calculations of Pollock, the penalty term for the kth function can also be expressed in matrix notation:

$$\text{Pen}(f_k, \lambda) = \frac{2}{3}\lambda \boldsymbol{\beta}_k^T \boldsymbol{R}_k \boldsymbol{\beta}_k = \frac{2}{3}\lambda \boldsymbol{\theta}_k^T \boldsymbol{Q}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{Q}_k \boldsymbol{\theta}_k.$$

For function k, let $\boldsymbol{J}_k$ once again be a binary matrix (size $n \times d_k$), where each row contains a single 1 indicating the knot to which sample i corresponds: $(\boldsymbol{J}_k)_{id} = 1 \Leftrightarrow X_i \boldsymbol{\xi}_k = t_d$. The

penalised residual sum of squares across all functions becomes

$$\begin{aligned}\mathrm{PSS}(\boldsymbol{f}, \lambda, \boldsymbol{Y}) &= \sum_i (Y_i - \textstyle\sum_k f_k(X_i \boldsymbol{\xi}_k))^2 + \sum_k \mathrm{Pen}(f_k, \lambda) \\ &= (\boldsymbol{Y} - \textstyle\sum_k \boldsymbol{J}_k \boldsymbol{\theta}_k)^T (\boldsymbol{Y} - \textstyle\sum_k \boldsymbol{J}_k \boldsymbol{\theta}_k) + \frac{2}{3}\lambda \textstyle\sum_k \boldsymbol{\theta}_k^T \boldsymbol{Q}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{Q}_k \boldsymbol{\theta}_k.\end{aligned}$$

If we wish to minimise this equation with respect to the kth function, we need only consider the marginal penalised residual sum of squares

$$\mathrm{PSS}(f_k, \lambda, \hat{\boldsymbol{Y}}) = (\hat{\boldsymbol{Y}} - \boldsymbol{J}_k \boldsymbol{\theta}_k)^T (\hat{\boldsymbol{Y}} - \boldsymbol{J}_k \boldsymbol{\theta}_k) + \frac{2}{3}\lambda \boldsymbol{\theta}_k^T \boldsymbol{Q}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{Q}_k \boldsymbol{\theta}_k,$$

where $\hat{\boldsymbol{Y}}$ represents the residuals, found by removing from $\boldsymbol{Y}$ the contributions of the remaining functions: $\hat{\boldsymbol{Y}} = \boldsymbol{Y} - \sum_{k' \neq k} \boldsymbol{J}_{k'} \boldsymbol{\theta}_{k'}$. To obtain the value of $\boldsymbol{\theta}_k$ which minimises this equation, we can differentiate and set to zero, obtaining

$$\boldsymbol{J}_k^T (\hat{\boldsymbol{Y}} - \boldsymbol{J}_k \boldsymbol{\theta}_k) = \frac{2}{3}\lambda \boldsymbol{Q}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{Q}_k \boldsymbol{\theta}_k.$$

This can be solved to find $\boldsymbol{\theta}_k$, and any other spline coefficients we desire. Notice that the penalty term depends only on the knot locations. Therefore, if $\boldsymbol{Y}$ is rescaled, the coefficients of $\boldsymbol{\theta}_k$ in the minimising spline will scale accordingly. However, if the knot values are rescaled, the coefficients of $\boldsymbol{\theta}$ in the minimising spline will change in an unpredictable fashion.

6.2.1 Bayesian Adaptation of Projection Pursuit Algorithm

Frequentist projection pursuit is not suitable for large numbers of predictors, so my original aim was to develop a sparse Bayesian version, one in which the contributions of most predictors to a projection are expected to be negligible. Inspired by the algorithm of ESL, I began with a hybrid version in which the functions were updated in a frequentist fashion and the directions sampled in a Bayesian manner. Unfortunately my implementation of this version performed poorly, but it did produce some useful conclusions.

I decided the decision of ESL to update direction-function pairs individually was necessitated by a lack of identifiability. If a constant is added to each component of $\boldsymbol{\theta}_k$, and subtracted from $\boldsymbol{\theta}_{k'}$, neither the smoothness of f_k nor $f_{k'}$ will be affected, nor the overall fit. For this reason, it is not possible to fit simultaneously more than one function. To restore identifiability, we can introduce a global intercept and set the last component of $\boldsymbol{\theta}_k$ to zero for each function. Consider the effect this has on the matrix notation $\boldsymbol{R}_k \boldsymbol{\beta}_k = \boldsymbol{Q}_k \boldsymbol{\theta}_k$. The left hand side remains unchanged. On the right hand side, the last column of $\boldsymbol{Q}_k$ becomes redundant. Therefore, let $\boldsymbol{Q}_k^-$ denote the matrix $\boldsymbol{Q}_k$ with the last column removed. In the same fashion, remove the last column from $\boldsymbol{J}_k$ to obtain $\boldsymbol{J}_k^-$ and the last element of $\boldsymbol{\theta}_k$ to obtain $\boldsymbol{\theta}_k^-$. To minimise

$\text{PSS}(\boldsymbol{f}, \lambda, \boldsymbol{Y})$ across all functions simultaneously, let $\boldsymbol{V}_k = \frac{2}{3}\boldsymbol{Q}_k^{-T}\boldsymbol{R}_k^{-1}\boldsymbol{Q}_k^{-}$ and construct the matrix $\boldsymbol{V}$ and vector $\boldsymbol{\Theta}$, where

$$\boldsymbol{V} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & \boldsymbol{V}_1 & 0 & \cdots & 0 \\ 0 & 0 & \boldsymbol{V}_2 & \cdots & 0 \\ 0 & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \boldsymbol{V}_K \end{bmatrix} \quad \text{and} \quad \boldsymbol{\Theta} = \begin{bmatrix} \theta_0 \\ \boldsymbol{\theta}_1^{-} \\ \boldsymbol{\theta}_2^{-} \\ \vdots \\ \boldsymbol{\theta}_K^{-} \end{bmatrix}$$

Then, with $\boldsymbol{J} = [\,\mathbf{1}\, \boldsymbol{J}_1^{-}\, \boldsymbol{J}_2^{-}\, \cdots\, \boldsymbol{J}_K^{-}\,]$, the penalised residual sum of squares can be written as

$$\text{PSS}(\boldsymbol{f}, \lambda, \boldsymbol{Y}) = (\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\Theta})^T(\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\Theta}) + \lambda\boldsymbol{\Theta}^T\boldsymbol{V}\boldsymbol{\Theta},$$

which is minimised when $\boldsymbol{J}^T(\boldsymbol{Y} - \boldsymbol{J}\boldsymbol{\Theta}) = \boldsymbol{V}\boldsymbol{\Theta}$. Now, given the directions $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \dots, \boldsymbol{\xi}_K$, which dictate the projections and hence the knots, the best $f_1, f_2, \dots, f_K$ can be computed explicitly. To me, updating multiple functions simultaneously seems far more pleasing than doing so individually, especially as we envisage projections having to "work together" to explain variance. Although the per-step computational demands would increase, I feel this would be more than compensated for by the increased efficiency of the algorithm.

Although I introduced this change in my implementation, there remained problems. The spike and slab priors I used for the direction coefficients had the effect of discretising the model space according to whether a predictor contributed to a particular projection or not. Suppose we consider whether to introduce predictor g to projection k by accepting a non-zero value of ξ_{kg}. When the prior probabilities of association are small, the penalty attached to such a move is very harsh. To have a reasonable chance of acceptance, the improvement in fit must be dramatic. However, to calculate the improvement in fit, the algorithm uses the current function f_k with the new direction $\boldsymbol{\xi}_k$. As f_k was determined with predictor g not involved, it is unlikely to be particularly suitable, leading to a very small probability of inclusion.

The harm of this discretisation might be less severe if instead a shrinkage prior was used. Nonetheless, I decided it was necessary to update directions and functions jointly. Therefore, when proposing ξ_{kg}, the algorithm calculated a revised set of functions $\boldsymbol{f}$ fitted to the new value. However, even with this change, I was unsatisfied with the method's performance. I believe an inherent problem was the notion of sampling from the joint model space of directions and functions, even though the primary aim was to make inferences only about the former. While taking this approach is valid in theory, having to worry about convergence of each f_k, as well as of each $\boldsymbol{\xi}_k$, must greatly reduce efficiency.

The last reason for abandoning this version was that I intended to use projection pursuit

for the detection of interactions between genetic variants, most of which take only two or three values. I realised that a spline based approach was unnecessarily complicated in this situation. This prompted me to discard the idea in favour of *Sparse Partitioning*, and it was only much later that I revisited the technique.

6.3 Bayesian Projection Pursuit

As the *Sparse Partitioning* methodology developed, I realised the similarities between that and a sparse version of projection pursuit. Primarily, the search for groups of associated predictors mimics the search for directions with only a few non-zero elements. In fact, my implementation of *Sparse Partitioning* already utilises projections; for each group of associated predictors, I first evaluate $\sum_j 3^{X_{\boldsymbol{G}_{kj}}}$, which maps each of the 3^{s_k} possible nodes to a distinct scalar value. When determining to which node each sample corresponds, it suffices to compare these projected values. This suggested that, instead of trying to design a Bayesian adaptation of the original projection pursuit algorithm, I might be able to incorporate its features directly into *Sparse Partitioning*. Although I view the resulting implementation more as an extension of *Sparse Partitioning* than a stand-alone method, to avoid confusion I refer to it as *Bayesian Projection Pursuit*.

The set-up for *Bayesian Projection Pursuit* is identical in almost all respects to *Sparse Partitioning*. The underlying relationship remains

$$f(\boldsymbol{X}) = f_1(X_{\boldsymbol{G}_1}) + f_2(X_{\boldsymbol{G}_2}) + \cdots + f_K(X_{\boldsymbol{G}_K}).$$

Now, the argument of the kth function is the projection of a sample's predictor values onto $\boldsymbol{\xi}_k = (\xi_{k1}, \xi_{k2}, \ldots, \xi_{kN})$, the kth direction: $f_k(X_{\boldsymbol{G}_k}) = f_k(\boldsymbol{X}\boldsymbol{\xi}_k)$. The group k determines which elements of the kth direction are nonzero: $g \in \boldsymbol{G}_k \Leftrightarrow \xi_{kg} \neq 0$. Each $\mathbb{G}$ continues to define a partitioning of the predictors, meaning that, as it stands, each predictor is only able to contribute to a single projection and thus be involved in a single function. To overcome this restriction, I retain the cheat of allowing multiple copies of predictors.

Let $\boldsymbol{\Xi} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_K\}$ represent the directions. In *Bayesian Projection Pursuit*, the exact model is represented by a triplet $\{\mathbb{G}, \boldsymbol{f}, \boldsymbol{\Xi}\}$. However, as before, $\boldsymbol{f}$ is a nuisance parameter, so we will be interested in the model space of pairs $\{\mathbb{G}, \boldsymbol{\Xi}\}$. Neither are we interested in the directions, but it does not prove possible to marginalise over their values. Notice that there is a large element of redundancy in this model representation; the partition determines the zero elements of the directions, while the directions determine the partitions entirely. It proves convenient to retain the partitions, but instead consider the parametrisation $\{\mathbb{G}, \boldsymbol{\Upsilon}\}$, where $\boldsymbol{\Upsilon} = (\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_N)$. The vector $\boldsymbol{\Upsilon}$ is analogous to the use of $\boldsymbol{I}$ to represent a par-

tition. If predictor g is associated, then Υ_g provides the corresponding direction coefficient: $g \notin \boldsymbol{G}_0 \Rightarrow \xi_{I_g g} = \Upsilon_g$. This use of $\boldsymbol{\Upsilon}$ is valid because each (copy of a) predictor can feature in no more than one projection.

6.3.1 Priors

The prior distribution takes the form $\mathbb{P}(\mathbb{G}, \boldsymbol{f}, \boldsymbol{\Upsilon}) = \mathbb{P}(\mathbb{G}) \times \mathbb{P}(\boldsymbol{\Upsilon}|\mathbb{G}) \times \mathbb{P}(\boldsymbol{f}|\mathbb{G}, \boldsymbol{\Upsilon})$. The partition prior remains the same, so it is necessary to decide a prior for $\boldsymbol{\Upsilon}$ and a revised prior for $\boldsymbol{f}$.

Direction Prior $\mathbb{P}(\boldsymbol{\Upsilon}|\mathbb{G})$

Firstly, we must decide a suitable domain for each direction coefficient. Suppose a group contains predictor g and we consider introducing predictor g'. The projection corresponding to this group, determined by Υ_g and $\Upsilon_{g'}$, will map the nodes $(X_g, X_{g'})$ to the real line. We would like there to be as much flexibility as possible in how the projection orders and locates the nodes. Therefore, it is appropriate that each direction coefficient's domain is $\mathbb{R}$. Saying this, a uniform prior across the real numbers would not be suitable. To be a meaningful projection, the coefficients must be of similar magnitude, otherwise the predictor corresponding to the largest coefficient would dominate the projection, and therefore the spline.

As it will not prove possible to integrate over coefficients, nor sample directly from their conditional posterior distributions, there is no pressure to choose a conjugate prior. Beginning with the case when predictor g is associated, I decided to assign Υ_g an independent normal distribution. The shape of this distribution is irrelevant, as shifting projection values will not affect the penalty term and nor the spline. Likewise, any misspecification in its scale can be compensated for by choice of λ (although similar to r, one might argue λ should be specific to each group). Therefore, I set each prior's mean to 0 and variance to 1:

$$\mathbb{P}(\Upsilon_g | g \notin \boldsymbol{G}_0) = \phi(\Upsilon_g).$$

When predictor g is not associated, it seems logical to force Υ_g to equal zero, which would induce a spike and slab marginal prior:

$$\mathbb{P}(\Upsilon_g) = \sum_{\mathbb{G}} \mathbb{P}(\Upsilon_g|\mathbb{G})\mathbb{P}(\mathbb{G}) = (1 - p_g)\delta_{\{0\}} + p_g \phi(\Upsilon_g).$$

However, when predictor g is not associated, the value of Υ_g becomes irrelevant. It will neither feature in any likelihood calculations nor be involved in any posterior estimates. Therefore, for reasons of convenience, which become clear later on, I assign a standard normal prior to each Υ_g, irrespective of the status of the gth predictor: $\mathbb{P}(\Upsilon_g) = \phi(\Upsilon_g)$.

Function Prior $\mathbb{P}(\boldsymbol{f}|\mathbb{G}, \boldsymbol{\Upsilon})$

As each spline is identifiable by its mapping of the observed knot values, a prior on the functions, $\mathbb{P}(\boldsymbol{f}|\mathbb{G}, \boldsymbol{\Upsilon})$, is equivalent to a prior on the coefficients, $\mathbb{P}(\boldsymbol{\Theta}|\mathbb{G}, \boldsymbol{\Upsilon})$. As there is often an equivalence between penalty terms and prior distributions, I use the integral of the second derivative as a starting point for devising a prior. Earlier, I expressed this integral as

$$\text{Pen}(\boldsymbol{f}, \lambda) = \frac{2}{3}\lambda \sum_k \boldsymbol{\theta}_k^T \boldsymbol{Q}_k^{-T} \boldsymbol{R}_k^{-1} \boldsymbol{Q}_k^{-} \boldsymbol{\theta}_k = \lambda \boldsymbol{\Theta}^T \boldsymbol{V} \boldsymbol{\Theta}.$$

This is reminiscent of (the negative logarithm of) a normal distribution with mean 0 and variance $\boldsymbol{V}^{-1}$. Though for this to be the case, $\boldsymbol{V}$ must be invertible, so in particular each $\boldsymbol{V}_k$ must have rank $d_k - 1$. However, $\boldsymbol{V}_k$'s rank is bounded by that of $\boldsymbol{Q}_k^{-}$, which itself has rank bounded by its smallest dimension, $d_k - 2$.

A first thought was to use the improper prior $\exp(-\boldsymbol{\Theta}^T \boldsymbol{V} \boldsymbol{\Theta})$, but this would be inappropriate as such a prior is not constant across all models. Therefore, I decided to increase the rank of $\boldsymbol{V}$ using a matrix transformation. I considered two possible transformations: $\boldsymbol{V} \to r\boldsymbol{I}_D + \boldsymbol{V}$ and $\boldsymbol{V} \to r\boldsymbol{1}_D' + \boldsymbol{V}$, where $\boldsymbol{1}_D'$ is the block diagonal matrix with diagonal components $1, \boldsymbol{1}_{d_1 \times d_1}, \boldsymbol{1}_{d_2 \times d_2}, \ldots, \boldsymbol{1}_{d_K \times d_K}$. Each transformation can be better understood by its effect on the penalty term corresponding to each function. The first adds to $\text{Pen}(f_k, \lambda)$ a penalty based on the sum of squares $\sum_d \theta_{kd}^2$, while the second adds a penalty based on the square of the sum $(\sum_d \theta_{kd})^2$.

In support of the first transformation, it seems reasonable for each function to have a penalty attached to the magnitude of its regression coefficients, although to some extent this is already provided for by $\boldsymbol{V}_k$. However, if either transformation is chosen purely out of necessity, I feel that a penalty based on $\sum_d \theta_{kd}$ would be the least disruptive. Lacking a clear-cut reason to pick one over the other, I opted for the first, primarily because of the elegance it afforded the model. The prior for the functions becomes

$$\mathbb{P}(\boldsymbol{f}|\mathbb{G}, \boldsymbol{\Upsilon}) = \mathbb{P}(\boldsymbol{\Theta}|\mathbb{G}, \boldsymbol{\Upsilon}) = (2\pi\sigma^2)^{-\frac{D}{2}} |r\boldsymbol{I}_D + \lambda \boldsymbol{V}|^{\frac{1}{2}} \exp\left\{-\tfrac{1}{2\sigma^2} \boldsymbol{\Theta}^T (r\boldsymbol{I}_D + \lambda \boldsymbol{V}) \boldsymbol{\Theta}\right\}.$$

Notice that this places an independent normal prior $\mathbb{N}(0, \sigma^2/r)$ on the intercept term. This prior can be viewed as a generalisation of that for categorical predictors; when $\lambda = 0$ the prior of *Sparse Partitioning* is recovered (demonstrated for the case when $S = 1$ by Biller and Fahrmeir, 1997). If elegance seems a poor argument on which to base a prior choice, then in any case the relative contributions of the two penalties can be altered by adjustment of r and λ. Saying that, very small r/λ should be avoided, as this might result in numerical errors as the inversion of $r\boldsymbol{I}_D + \lambda \boldsymbol{V}$ becomes less stable. If r is set to zero, the method will run, but once

again using the semi-frequentist version which replaces marginal likelihoods with maximum likelihoods.

At the moment, λ remains constant throughout the analysis, assigned a default value of 1. The user is able to specify a value in advance, according to how much smoothing they desire (a small smoothing parameter indicates moderate smoothing, a big one indicates large smoothing). It would, naturally, be preferable to assign λ a prior, which would make the method more robust to misspecification. It has been suggested to me that this might be possible. However, at the moment, I have been unable to envisage such an implementation, so the next best approach is to run the method using different choices, and compare the results for each.

6.3.2 Likelihood

With the addition of $\boldsymbol{\Upsilon}$, the raw likelihood equations take the form $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}, \boldsymbol{\Theta}, \sigma^2)$ or $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}, \boldsymbol{\Theta})$, depending on whether the response is continuous or binary. As before, we are more interested in the marginal likelihood.

Marginal Likelihood

In Case 1 (continuous response, identity link function) and Case 2 (binary response, logit link function), we wish to calculate $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon})$. In Case 3 (binary response, probit link function), we are interested in $\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon})$. These marginal likelihoods are calculated in a similar fashion to their equivalent forms in *Sparse Partitioning*. When the response is continuous we obtain

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}) = |r\boldsymbol{I}_D + \lambda \boldsymbol{V}|^{\frac{1}{2}} (2\pi)^{-\frac{n}{2}} \times |\boldsymbol{B}|^{-\frac{1}{2}} \times \Gamma(\tfrac{n}{2}) (\boldsymbol{Y}^T\boldsymbol{Y} - \boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})^{-\frac{n}{2}},$$

where this time $\boldsymbol{B} = \boldsymbol{J}^T\boldsymbol{J} + r\boldsymbol{I}_D + \lambda\boldsymbol{V}$ and $\boldsymbol{A} = \boldsymbol{B}^{-1}\boldsymbol{J}^T\boldsymbol{Y}$.

When the response is binary and a logit link function is used, the marginal likelihood can again be calculated using a Laplace approximation.

$$\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}) \approx \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}, \hat{\boldsymbol{\Theta}}) \times \mathbb{P}(\hat{\boldsymbol{\Theta}}|\mathbb{G}, \boldsymbol{\Upsilon}) (2\pi)^{\frac{D}{2}} \left| -\frac{\mathrm{d}^2 w(\hat{\boldsymbol{\Theta}})}{\mathrm{d}\boldsymbol{\Theta}^2} \right|^{-\frac{1}{2}},$$

where, like before, $w(\boldsymbol{\Theta})$ is the logarithm of $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}, \boldsymbol{\Theta}) \times \mathbb{P}(\boldsymbol{\Theta}|\mathbb{G}, \boldsymbol{\Upsilon})$, the integrand of the marginal likelihood, and $\hat{\boldsymbol{\Theta}}$ is the value which maximises $w(\boldsymbol{\Theta})$. The first and second derivative matrices of $w(\boldsymbol{\Theta})$, required for the equation above and to estimate the posterior mode of $\boldsymbol{\Theta}$, can be calculated in the same manner as before, simply by replacing the (inverse)

variance matrix $r\boldsymbol{I}_D$ by $r\boldsymbol{I}_D + \lambda\boldsymbol{V}$. This time, we find that

$$w(\boldsymbol{\Theta}) = \sum_i Y_i \log p_i + (1 - Y_i)\log(1 - p_i) - \frac{1}{2}\boldsymbol{\Theta}^T(r\boldsymbol{I}_D + \lambda\boldsymbol{V})\boldsymbol{\Theta} - \frac{D}{2}\log(2\pi) + \frac{1}{2}\log(|r\boldsymbol{I}_D + \lambda\boldsymbol{V}|),$$

so

$$\frac{\mathrm{d}}{\mathrm{d}\Theta_j} w(\boldsymbol{\Theta}) = \sum_i (Y_i - p_i)J_{ij} - r\Theta_j - \lambda\boldsymbol{V}_j\boldsymbol{\Theta}$$

and

$$\frac{\mathrm{d}^2}{\mathrm{d}\Theta_j \mathrm{d}\Theta_k} w(\boldsymbol{\Theta}) = \sum_i -p_i(1 - p_i)J_{ij}J_{ik} - r\mathbf{1}(j = k) - \lambda\boldsymbol{V}_{jk}.$$

These equations can be substantially simplified because of the sparse nature of $\boldsymbol{V}$. For example, in the first derivative, suppose $\Theta_j = \theta_{kd}$ a regression coefficient corresponding to the kth group. The jth row of $\boldsymbol{V}$ will be nonzero only for elements relating to group k, so $\boldsymbol{V}_j\boldsymbol{\Theta}$ reduces to $(\boldsymbol{V}_k)_d\boldsymbol{\theta}_k^-$.

When the response is binary and a probit link function is used, the marginal likelihood is the same as for the continuous response case with $\boldsymbol{Y}$ replaced by $\boldsymbol{Z}$ and variance σ^2 set to 1:

$$\mathbb{P}(\boldsymbol{Z}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}) = |r\boldsymbol{I}_D + \lambda\boldsymbol{V}|^{\frac{1}{2}}(2\pi)^{-\frac{n}{2}} \times |\boldsymbol{B}|^{-\frac{1}{2}} \times \exp\{-\tfrac{1}{2}(\boldsymbol{Z}^T\boldsymbol{Z} - \boldsymbol{A}^T\boldsymbol{B}\boldsymbol{A})\},$$

with $\boldsymbol{A}$ and $\boldsymbol{B}$ as just defined.

6.3.3 Posterior Distribution

Depending on the link function, the aim is to sample from $\mathbb{P}(\mathbb{G}, \boldsymbol{\Upsilon}|\boldsymbol{X}, \boldsymbol{Y})$ or $\mathbb{P}(\mathbb{G}, \boldsymbol{\Upsilon}, \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{Y})$. The presence of $\boldsymbol{\Upsilon}$ in the posterior distribution is unfortunate, but unavoidable. It is a matter of discussion how important its value and, therefore, how much it slows down convergence. Despite replacing $\boldsymbol{\Xi}$ with $\boldsymbol{\Upsilon}$, there remains a strong dependency between the partitions and $\boldsymbol{\Upsilon}$, which must be taken into consideration in the sampling stages. For example, it is pointless to propose a non-zero value of I_g if Υ_g equals zero. To avoid having to propose from a joint distribution, I do not insist that Υ_g equals zero when predictor g is not associated, which earlier allowed me to make its prior independent of I_g.

In order to explore values of $\boldsymbol{\Upsilon}$, *Bayesian Projection Pursuit* retains Sampling Stages One, Two and Three, and introduces Sampling Stage Four. As in *Sparse Partitioning*, it is frequently necessary to score partitions, however, this time the score is conditional on the

current value of $\boldsymbol{\Upsilon}$:

$$\text{Score}(\mathbb{G}|\boldsymbol{\Upsilon}) := \begin{cases} \mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}) \times \mathbb{P}(\mathbb{G}) \times \mathbb{P}(\boldsymbol{\Upsilon}), & \text{for Cases 1 and 2,} \\ \mathbb{P}(\boldsymbol{Z}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon}) \times \mathbb{P}(\mathbb{G}) \times \mathbb{P}(\boldsymbol{\Upsilon}), & \text{for Case 3.} \end{cases}$$

Like before, I describe Sampling Stages One, Two and Four for Cases 1 and 2. For Case 3, the instructions remain correct provided $\boldsymbol{Y}$ is replaced with $\boldsymbol{Z}$, which is simply achieved by using the corresponding scoring function.

Stage One: Sampling each Component of $\boldsymbol{I}$

I_g determines to which group the gth predictor belongs, or, alternatively, to which, if any, projection the predictor contributes. For each predictor in turn, a new value I_g^* is sampled from its conditional posterior distribution:

$$\mathbb{Q}(I_g^*) = \mathbb{P}(I_g^*|I_{-g}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Upsilon}) = \frac{\text{Score}(I_g^*, I_{-g}|\boldsymbol{\Upsilon})}{\sum_{I_g'} \text{Score}(I_g', I_{-g}|\boldsymbol{\Upsilon})}.$$

Stage Two: Sampling a Component of $\mathbb{G}$

A component from a non-null group of the partition is picked at random and a new value G_{kg}^* is sampled from its conditional posterior distribution:

$$\mathbb{Q}(G_{kg}^*) = \mathbb{P}(G_{kg}^*|\mathbb{G}_{-kg}, \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\Upsilon}) = \frac{\text{Score}(G_{kg}^*, \mathbb{G}_{-kg}|\boldsymbol{\Upsilon})}{\sum_{G_{kg}'} \text{Score}(G_{kj}', \mathbb{G}_{-kg}|\boldsymbol{\Upsilon})},$$

where, once again, $\mathbb{G}_{-kg}$ is shorthand to represent the current partition with element G_{kg} removed.

Stage Three: Sampling each Component of $\boldsymbol{Z}$

This stage is only applicable for Case 3, when a probit link function is used with a binary response. For each sample, a new value for the latent variable Z_i^* is proposed from a folded standard normal distribution, with sign determined by Y_i, and accepted with probability $\min(1, \alpha)$, where

$$\alpha = \frac{\mathbb{P}(Z_i^*, Z_{-i}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon})}{\mathbb{P}(Z_i, Z_{-i}|\boldsymbol{X}, \mathbb{G}, \boldsymbol{\Upsilon})} \times \frac{\phi(Z_i)}{\phi(Z_i^*)}.$$

Again, this is convenient to calculate, as the first fraction is the ratio of the scores for the current partition using the proposed and current values of Z_i.

Stage 4: Sampling each Component of $\boldsymbol{\Upsilon}$

For each predictor, a new value for Υ_g is proposed from its prior: $\mathbb{Q}(\Upsilon_g) = \phi(\Upsilon_g)$. The proposed value Υ_g^* is accepted with probability $\min(1, \alpha)$, where

$$\alpha = \frac{\mathbb{P}(\mathbb{G}, \Upsilon_g^*, \Upsilon_{-g} | \boldsymbol{X}, \boldsymbol{Y})}{\mathbb{P}(\mathbb{G}, \Upsilon_g, \Upsilon_{-g} | \boldsymbol{X}, \boldsymbol{Y})} \times \frac{\phi(\Upsilon_g)}{\phi(\Upsilon_g^*)}.$$

With rearrangement, this becomes

$$\alpha = \frac{\mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}, \mathbb{G}, \Upsilon_g^*, \Upsilon_{-g}) \times \mathbb{P}(\mathbb{G}) \times \mathbb{P}(\Upsilon_g^*, \Upsilon_{-g})}{\mathbb{P}(\boldsymbol{Y} | \boldsymbol{X}, \mathbb{G}, \Upsilon_g, \Upsilon_{-g}) \times \mathbb{P}(\mathbb{G}) \times \mathbb{P}(\Upsilon_g, \Upsilon_{-g})} \times \frac{\phi(\Upsilon_g)}{\phi(\Upsilon_g^*)}.$$

When predictor g is not associated, the marginal likelihood does not depend on Υ_g, so the fractions cancel to 1 and the proposed value is accepted immediately. When predictor g is associated, the first fraction is the ratio of scores of the current model calculated given the proposed and current values of $\boldsymbol{\Upsilon}$.

For the first three sampling stages, the fix to allow multiple copies of predictors ($C > 1$) is the same as in the original version of *Sparse Partitioning* and has minimal effect on the processing time compared to $C = 1$. For Stage 4, most samplings will continue to be draws from a standard normal distribution, so the time remains negligible compared to the other more computationally intensive proposal steps.

Preprocessing and Additional Features

In addition to standardising continuous response values, *Bayesian Projection Pursuit* also standardises the predictor values. This serves two useful purposes. Firstly, it equalises their variances, making the constant choices of λ and r more appropriate. Secondly, it is beneficial when applied to tertiary predictors. This increases the flexibility of the model, as it removes the state 0, which would otherwise not (directly) contribute to any projections.

Bayesian Projection Pursuit offers the range of extra features available in *Sparse Partitioning*. As the direction coefficients are constant outside of Sampling Stage Four, the steps for sampling missing states, predicting response values and adjusting for confounding are performed in an identical fashion.

Discussion: Importance of $\boldsymbol{\Upsilon}$

For the sampling stages, it proved very convenient to relax the condition that Υ_g is set to 0 whenever predictor g is not associated. It removes the need to jointly sample a new direction coefficients and a partition. As the joint conditional posterior distribution of, say, Υ_g and I_g

would not be tractable (else the directions could be marginalised over), Sampling Stages One and Two would have to resort to Metropolis-Hastings sampling and, I believe, be less efficient.

In most situations, relaxing this condition is of no consequence. When the algorithm considers declaring a predictor associated, it would already have a direction coefficient ready to use. If this coefficient proves unsuitable, a new one will be available for the next iteration. In effect, this is equivalent to proposing a new coefficient at the same time as proposing a predictor becomes associated. When the algorithm considers removing an associated predictor, the value of its direction coefficient becomes irrelevant, so a joint proposal is not required. The only danger situation occurs when considering whether to move an associated predictor between non-null groups. The current direction coefficient will have been sampled conditional on the predictor being in its current group, which might not necessarily be a good choice if the predictor moves to a new group. Therefore, in this situation, it might be advisable for me to consider proposing jointly a new direction coefficient along with the new group membership.

The penalty term $\text{Pen}(\boldsymbol{f}, \lambda)$ — which inspired the prior distribution for functions, which in turn impacts each model's score — is affected by the knot locations of each projection, and therefore by the set of direction coefficients $\Upsilon_{S_{\boldsymbol{I}}}$. Ideally, the score for a partition would be an integral across all $\boldsymbol{\Upsilon}$. Although this would only be required for Υ_g, the projection values corresponding to the associated predictors, this would still not be possible. Crucial to the marginal likelihood, are the relative values of elements of $\boldsymbol{\Upsilon}$, which determine the spacing and ordering of knots. Saying this, unfortunately the penalty term is not invariant to rescaling all projection values. For example, consider the simple case, when $\boldsymbol{G}_k$ is a singleton group containing just predictor g. The projected values $\boldsymbol{X\xi}_k = X_g \xi_{kg}$ will always have the same relative spacings, regardless of the choice of ξ_{kg}. However, if the direction coefficient is multiplied by c, the penalty term will be divided by c^3. Although this is as expected — more spaced-out knots allow the function to "wiggle" more in the direction of the y-axis for the same penalty — this property makes the direction values more important.

Nonetheless, I believe there is little harm in standardising the knots / projection values. Although it will affect the prior over the functions, it will not take away their generality nor, most importantly, alter the prior probabilities of association. Therefore, before calculating the matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$, I linearly transform the knots so that the first and last values are at 0 and $n - 1$. Notice that the signs of the knots still play a minor role, as they determines the choice of each spline's base knot, for which the regression coefficient is set to zero.

This mapping has additional advantages. When two knots are very close together, the inversions, in particular that involved in calculating $\boldsymbol{R}_k^{-1}\boldsymbol{Q}_k^{-}$, can become unstable. Standardising knots alleviates this concern to some extent. Furthermore, as I have described the

algorithm, over 95% of the computation time is spent calculating the score. Of this, over 75% is spent sorting the projection values into order to deduce the knots. Ironically, this process is slower for tertiary predictors, as when there are only a few knots, the sort algorithm repeatedly swaps identical values back and forth. Standardising the knots suggested a speed-up. If each projection is rounded down to the nearest integer, the task of finding the location of each knot, and to which samples it corresponds, becomes much faster. It also provides a certain elegance, as the code becomes the same as that for the original version of *Sparse Partitioning*. If this approximation was a concern, we could increase the range of transformed projections by a few factors, which would increase accuracy, but still maintain a speed-up.

Reducing Computation Time

Computation time is an issue with *Bayesian Projection Pursuit*. Even if the process of sorting projection values into order was instantaneous, the algorithm would still take much longer than an equivalent run of *Sparse Partitioning*. The run time would primarily depend on the degrees of freedom of each model, as this determines the dimensions of $\boldsymbol{R}_k$ and $\boldsymbol{B}$, two matrices which must be inverted. In *Sparse Partitioning*, a single function will have maximum degrees of freedom 3^{s_k}, so perhaps 9 or 27; in *Bayesian Projection Pursuit*, each spline will have degrees of freedom equal to the number of distinct projections, bounded only by n. Therefore, not only does it solve the sorting problem, the approximation of transforming each observed value to the nearest integer can also greatly reduce the degrees of freedom. Later on, I present a toy example. For this, allowing knot values to be approximated by integers speeds up run-time by a factor of five, with no discernible effect on performance. In fact, I take this approximation further. I transform the projected values onto even smaller intervals, while still forcing each to an integer, and observe faster run times with no noticeable effect on accuracy.

I feel it is possible to justify this approximation. Primarily, a drop in accuracy arises when the transformation maps distinct projection values to the same integer, so reducing the number of knots and thus the (absolute) degrees of freedom of each function. However, of more importance is the "effective degrees of freedom". This can formally be defined as the trace of the "hat" matrix. One of the final steps in calculating a best fit is to compute the maximum likelihood estimate or posterior mode of the regression coefficients. For linear models, this will be a linear transformation of the response values, taking the form $\boldsymbol{B}^{-1}\boldsymbol{J}^T\boldsymbol{Y}$. To calculate the fitted response values, we would premultiply this value by $\boldsymbol{J}$, providing us with $\hat{\boldsymbol{Y}} = (\boldsymbol{J}\boldsymbol{B}^{-1}\boldsymbol{J}^T)\boldsymbol{Y}$. Again, this is a linear transformation of the response $\boldsymbol{Y}$. The term within the parentheses is called the hat matrix.

The impact of the penalty term / prior distribution is to reduce the effective degrees of freedom from D. This process happens most rapidly for higher values. Therefore, even if the

approximation cuts the number of knots quite dramatically, the net reduction of the effective degrees of freedom is far less severe. Nonetheless, I intend to study more formal routes for reducing the degrees of freedom. For example, B-splines are a commonly used option (Eilers and Marx, 1996). These are formed from an unlimited set of basis functions. Unlike a natural cubic spline, which requires a degree of freedom per knot, B-splines allow the user to choose the degrees of freedom by deciding how many basis functions to use. For many problems, it is generally considered unnecessary to have more than, say, 20 degrees of freedom (Hastie *et al.*, 2001).

Alternatively, I could consider removing the splines altogether. Although they proved the most natural way to extend *Sparse Partitioning* for use with continuous predictors, with *Bayesian Projection Pursuit* my work has drifted towards the areas of feature selection and machine learning, suggesting I might be able to incorporate techniques from these fields. One solution might involve Gaussian processes (described with a genetic application in Chu *et al.*, 2005). Set up using my notation, the natural application would treat $f_k(X_{1G_k}), f_k(X_{2G_k}), \ldots, f_k(X_{nG_k})$ as realisations from a Gaussian process, distributed multivariate normally, with covariance matrix $\boldsymbol{V}_k$. The predictors X_{G_k} would influence this Gaussian process through the covariance matrix. For example, one possible construction is $(\boldsymbol{V}_k)_{ii'} = \sum_j \kappa_j X_{iG_{kj}} X_{i'G_{kj}}$. Here, the coefficient κ_j denotes the relative importance of predictor G_{kj} to this process, and would take the place of the corresponding direction coefficient in $\boldsymbol{\xi}$. Were $\boldsymbol{V}_k$ equal to the identity matrix, the process would have degrees of freedom n for each function, allowing complete flexibility (just like the case $\lambda = 0$). Instead, the construction of $\boldsymbol{V}_k$ reduces the effective degrees of freedom, similar to the introduction of splines. However, compared to splines, Gaussian processes require less computation ($\boldsymbol{V}$ can be computed readily, rather than first computing $\boldsymbol{Q}_k$ and $\boldsymbol{R}$, then inverting the latter), while they also remove the need for an explicit smoothing parameter. Saying that, it would remain to see whether this simplification comes at the expense of flexibility and, in particular, how well the implied smoothing could be tailored to the user's needs.

6.4 Simulated Data

In this section, I briefly test *Bayesian Projection Pursuit* in the same manner as in Chapter 5. I compare the method to *Sparse Partitioning*, *Deterministic SP*, *Pairs*, *MARS* and *best other*, which once again represents the best combined performance of the remaining five methods (*Single*, *CART*, *RF*, *SSS* and *Logic*). For these studies, *Bayesian Projection Pursuit* generally took about twice as long as *Sparse Partitioning* to run for the same number of iterations (200). However, as the datasets were relatively small (typically $n = 100$, $N = 1000$), each run still completed within a couple of minutes.

Figures 6.1 and 6.2 provide the results for Studies One to Six, as labelled in their captions. In every case except one, the performance of *Bayesian Projection Pursuit* tracked between *Sparse Partitioning* and the other methods. When the difference between methods was small, *Bayesian Projection Pursuit*'s performance was almost indistinguishable; when large, it exploited the gap to offer some improvement over the other methods. The only curiosity concerns Study Six, which considered binary response values. Once again, the dashed line indicates the performance when a probit link function was used. For the equivalent plot in Chapter 4, the probit option performed better, in part because it afforded the user more iterations. Here, this is not the case, suggesting the extra layer of complexity, brought about by the addition of $\boldsymbol{Z}$, causes more trouble than it is worth.

Based on these simulation studies, I conclude that *Bayesian Projection Pursuit* shows promise as a method. Although beaten by *Sparse Partitioning*, it has generally matched or outperformed all other methods. This is despite the fact that it is designed with continuous predictors in mind, the opposite end of the scale to the datasets tested here.

When λ is set to 0, the method runs identically to *Sparse Partitioning*. The directions remain in the model and technically contribute to the posterior score, but will have no effect on any of the first three sampling stages. In the simulation studies, I left λ at its default value of 1. Therefore, that *Bayesian Projection Pursuit* slightly underperformed *Sparse Partitioning* when applied to tertiary predictors, must stem from the introduction of splines. The effect of these is to reduce the effective degrees of freedom. For *Sparse Partitioning* the effective degrees of freedom is reduced from D only by the presence of priors for $\boldsymbol{\Theta}$ and σ^2. With *Bayesian Projection Pursuit*, it is additionally affected by the explicit smoothness penalty. However, that the method only performed slightly worse, once again backs up the idea that, in a Bayesian set-up, excessive generality, which for these studies, *Bayesian Projection Pursuit* very certainly possesses, does not necessarily come at the expense of performance.

Bayesian Projection Pursuit relies on the direction coefficients to lay the foundation for the splines. As I mentioned, were one direction to become much smaller than the others, its effect would be dwarfed in the model. The predictor will essentially become redundant, except to penalise the model for its inclusion. Therefore, the method requires a sensible choice of directions. Certainly the ordering is also important, very much so for binary predictors. Consider the multiplicative interaction contained in the underlying relationship of Model II. In *Bayesian Projection Pursuit*, the most harmonious solution, that which permits the smoothest spline, places the projection corresponding to (1,1) at one end of the knots. Otherwise, a correct mapping would require the function curve to double back on itself.

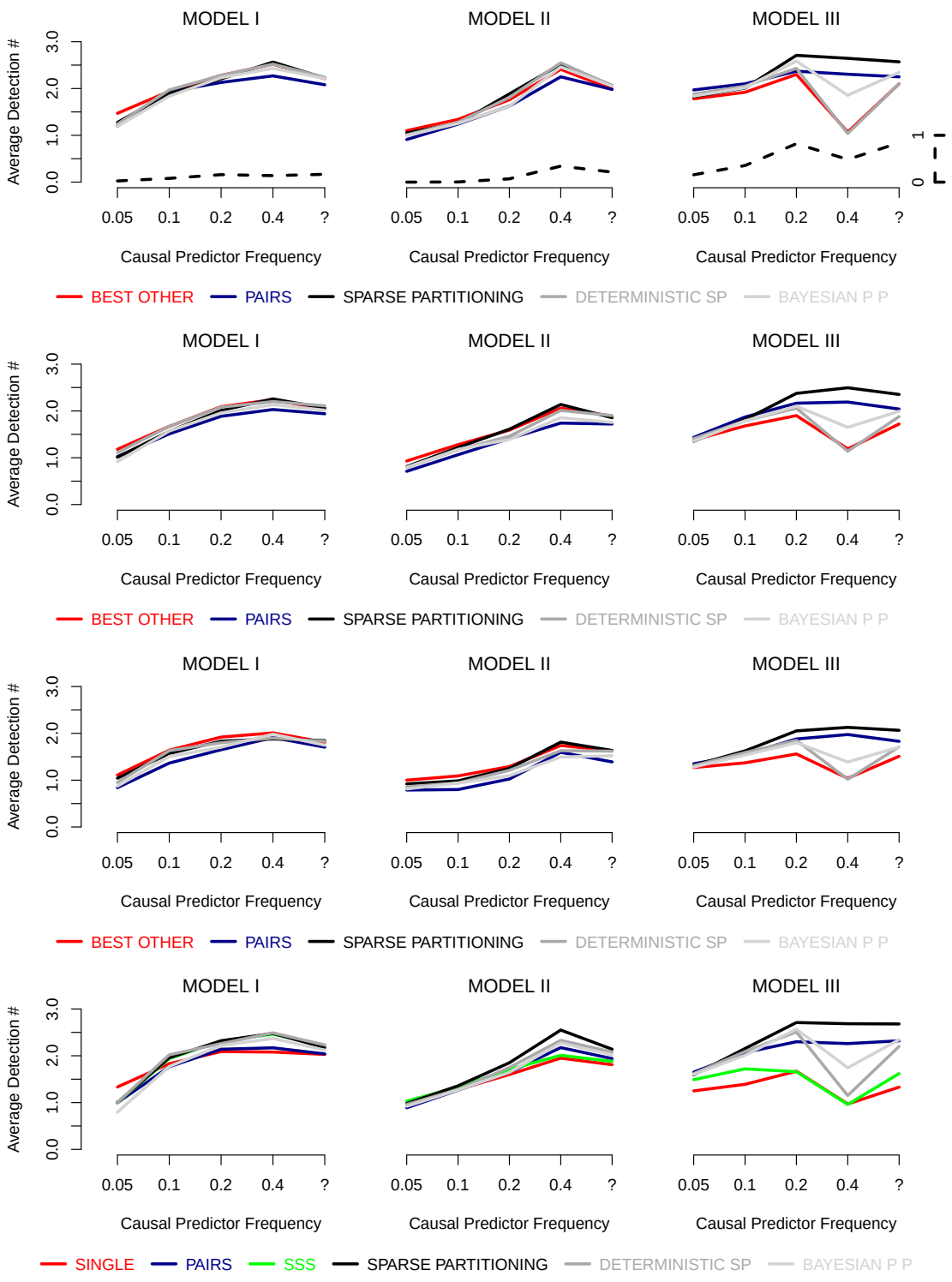


Figure 6.1: *Results of Simulation Studies One to Three. The top row corresponds to Study One (continuous response, idealised data), the middle two rows to Study Two (causal predictors unobserved, $r^2 = 0.9$ or $r^2 = 0.8$), the bottom row corresponds to Study Three (10% missing predictors).*

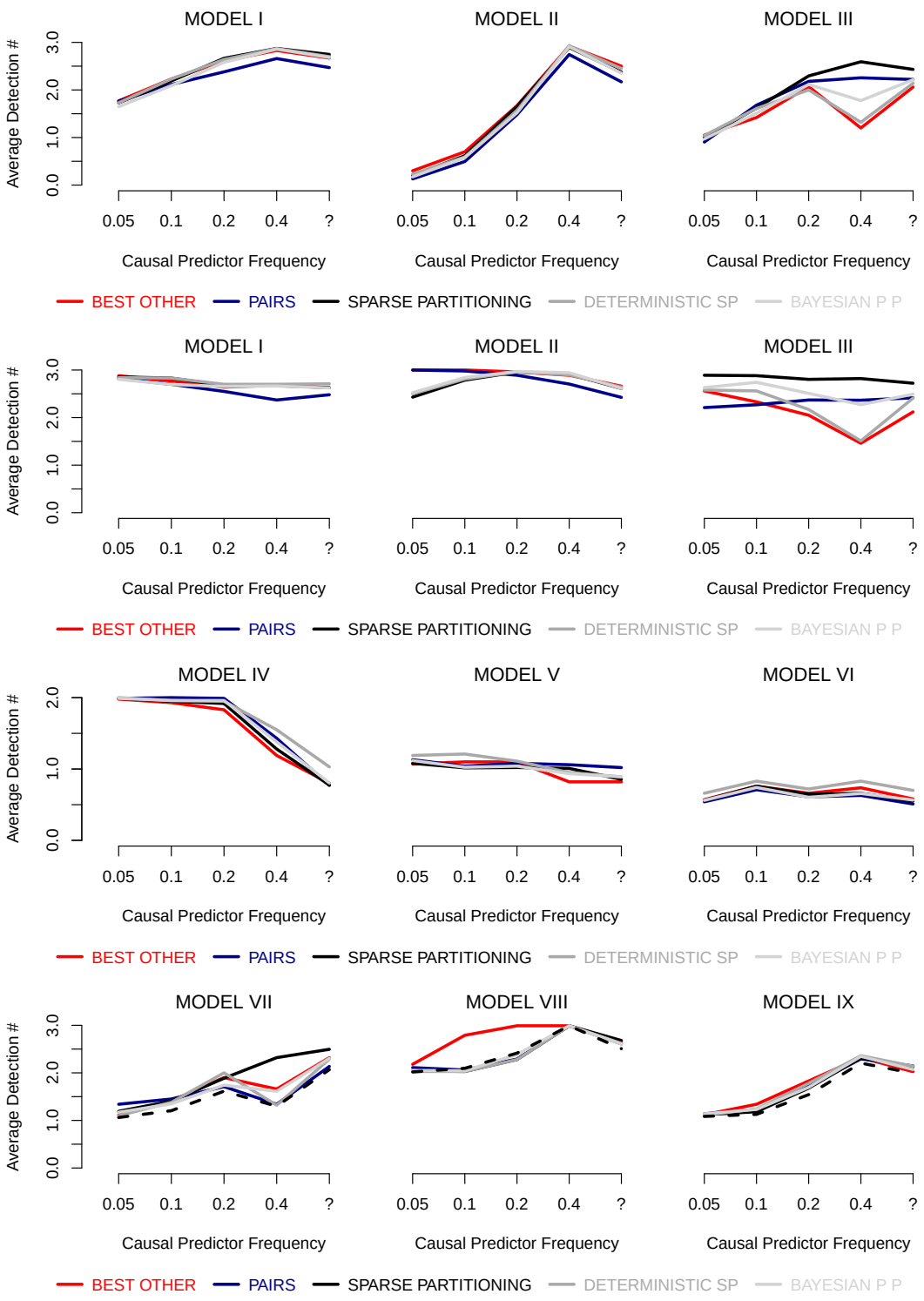


Figure 6.2: *Results of Simulation Studies Four to Six. The top two rows correspond to Study Four (noise distributed exponentially, then uniformly), the third row to Study Five (tertiary predictors), the bottom row to Study Six (binary response).*

Beyond this, I have not decided how important the actual direction values are. At the moment, a deterministic implementation of *Bayesian Projection Pursuit* is not possible because the inclusion of direction terms prevents calculation of the required marginal likelihood $\mathbb{P}(\boldsymbol{Y}|\boldsymbol{X}, \mathbb{G})$. However, the fact that there are clearly a lot of redundancies in the choice of directions, due to symmetry and the manner in which they are standardised, suggests an approximate version might be possible. It might be reasonable to score each partition for a small number of direction values which adequately approximate its effect over a range of functions. This is certainly the case when a non-null group is singleton, as for this the direction value becomes redundant. In a similar fashion, I sense a reasonable number of directions could well model a bivariate function. This idea is certainly one I would like to think more about.

6.5 Real Data

In this section, I show the results of applying *Bayesian Projection Pursuit* to a toy dataset. I am very grateful to Hastie *et al.*, the authors of ESL, for making their datasets available and explaining fully their methods. For my application, I consider the book's second example dataset, which concerns a study of prostate cancer (Stamey *et al.*, 1989). Here, the response (`lpsa`) measured the logarithm of prostate specific antigen, while the predictors recorded log cancer volume (`lcavol`), log prostate weight (`lweight`), age (`age`), log of benign prostatic hyperplasia amount (`lbph`), seminal vesicle invasion (`svi`), log of capsular penetration (`lcp`), "Gleason score" (`gleason`) and percent of "Gleason scores 4 or 5" (`pgg45`). ESL's aim was to see how well the eight clinical factors could predict this value. I have taken a similar approach. Although the number of predictors was small, this dataset provided a range of challenges. In particular, while six of the measurements were continuous, `svi` was binary valued and `gleason` was recorded in only 4 states. Therefore, for a method to be successful, it must be able to accommodate the varying nature of each clinical factor.

ESL compared the predictive performance of six frequentist methods: least squares regression (LSR), best subset selection (BSS), ridge regression (RR), the Lasso, principle component regression (PCR) and partial least squares regression (PLS). The first four methods, I have explained already. PCR performs standard least squares regression on one or more of the top principal component axes. These axes will be orthogonal, which could prove useful in the face of correlated raw predictors, which the ones in this study turn out to be. PLS regression similarly forms orthogonal linear combinations of predictors, but while PCR calculates these from only the predictors themselves, PLS is guided also by the response values. The weightings used to construct each combination are influenced by the strength of association between the response and each predictor in turn.

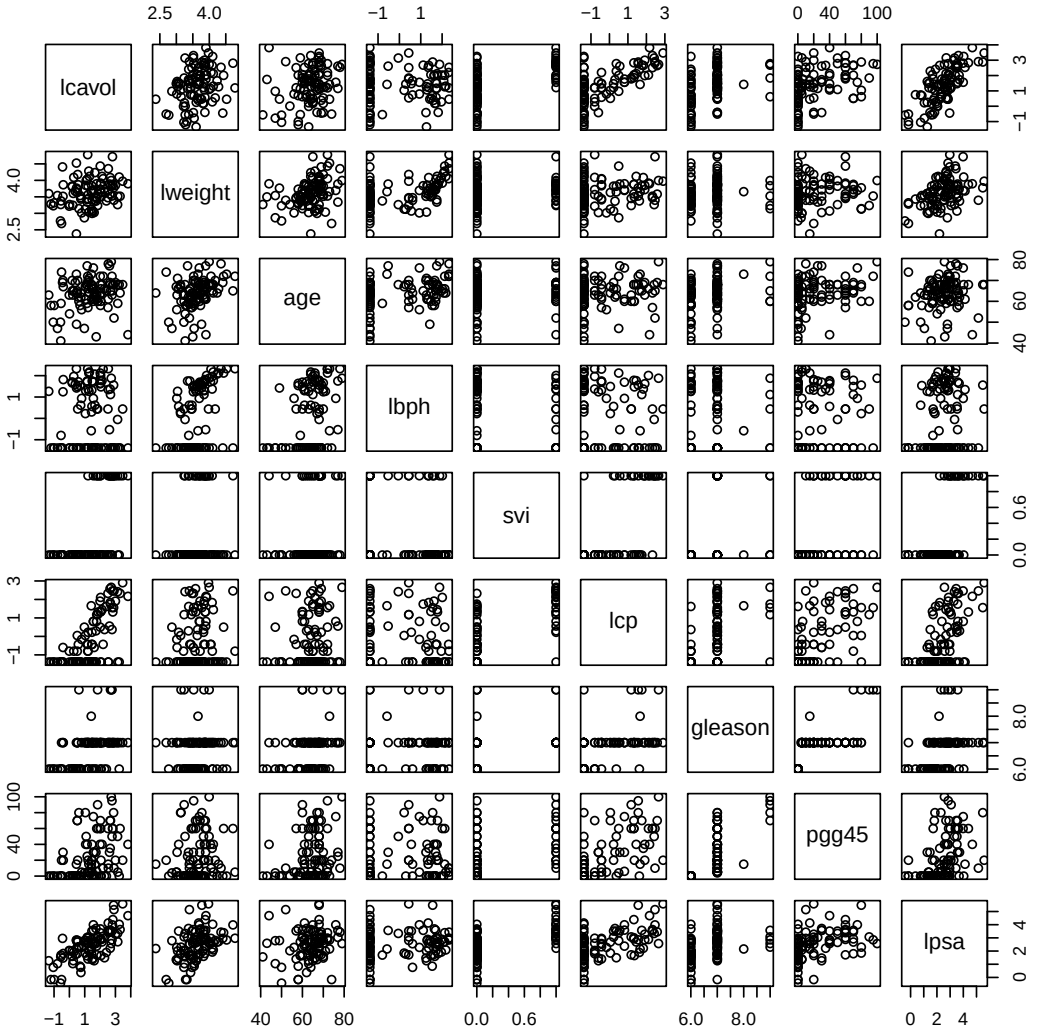


Figure 6.3: *Variables for the prostate cancer dataset examined in ESL (*Hastie et al.*, 2001). In total, there were 97 individuals measured for eight predictors and one response. Each cell represents a pairwise scatter-plot for two variables; the first eight rows / columns correspond to the predictor values, the ninth to the response. In particular, notice the correlations present; for example, between predictors 1 and 6. Furthermore, while most predictors were continuous valued, predictors 5 and 7 were discrete, taking either 2 or 4 values, respectively.*

Term	LSR	BSS	RR	Lasso	PCR	PLS
`Intercept`	2.480	2.495	2.467	2.477	2.513	2.452
`lcavol`	0.680	0.740	0.389	0.545	0.544	0.440
`lweight`	0.305	0.367	0.238	0.237	0.337	0.351
`age`	-0.141		-0.029		-0.152	-0.017
`lbph`	0.210		0.159	0.098	0.213	0.248
`svi`	0.305		0.217	0.165	0.315	0.252
`lcp`	-0.288		0.026		-0.053	0.078
`gleason`	-0.021		0.042		0.230	0.003
`pgg45`	0.267		0.123	0.059	-0.053	0.080
Prediction Error	0.586	0.574	0.540	0.491	0.527	0.636

Figure 6.4: *Final prediction models. The columns provide the 9 regression coefficients (the intercept term plus a coefficient per clinical factor) for the six frequentist methods. The clinical factors and method abbreviations are explained in the main text. Spaces indicate the corresponding variable did not contribute in the method's final model. The final row provides the prediction error for each method; the Lasso has performed best, while partial least squares regression has performed worst. Table reproduced from* The Elements of Statistical Learning *(*Hastie et al.*, 2001).*

In ESL, two stages of cross-validation were performed. To begin with, the 97 individuals were divided into training and test sets of 67 and 30 samples, respectively. Ultimately, each method was scored by recording mean squared prediction error for the test sample's response values. With the exception of LSR, each method requires an element of tuning. To do this, ten-fold cross-validation was performed on the training set. This entailed dividing the training samples into ten groups, taking turns to fit the data to nine of these groups, while assessing performance on the tenth. For each of these steps, the method was applied for a range of parameter choices and prediction score recorded. By averaging performance over the ten applications, it was possible to judge which parameter settings performed best. These settings were then carried over to the final analysis on the test dataset, from which the final assessment of model performance was obtained.

Figure 6.4 reproduces the details of the final prediction models for the six frequentist methods, as provided by ESL. All of these methods are linear, so their final models can be specified by an intercept term and the regression coefficient for each predictor (blank cells in a column indicate clinical factors which were not used in the corresponding method's final prediction model). The bottom row reports the mean squared prediction error for each method, the statistic used for comparison. Out of these models, the Lasso has fared best, so became the method to beat.

The large black squares in Figure 6.5, whose positions are constant across plots, indicate the performance of the six frequentist methods. The x-axis value corresponds to prediction error for the training samples, the y-axis value to prediction error for the test samples. Therefore, for *Bayesian Projection Pursuit* to outperform a particular method, its prediction error must lie below the corresponding horizontal line.

The smaller points in each plot represent the results from multiple runs of *Bayesian Projection Pursuit*. The three most important parameter settings in my method are r, p_g and λ, so I considered the effect of varying each. The plots in the left column correspond to setting $r = 0.5$, those in the right column correspond to setting $r = 1$. Each row relates to a different prior mean number of associations, selected from 0.5, 1 and 1.5 (i.e. p_g equal to 1/16, 2/16 or 3/16). Finally, for each combination of r and p_g, I tested five different values for λ, ranging from 0.1 to 1000 (low to high smoothing), as indicated by colour.

I also considered the impact of transforming the projected values to increasingly tight intervals. In total, I performed four sets of runs: the first set was the least approximate, obtaining the knots by first mapping the projected values to the interval [0,67), then rounding down to the nearest integer; the remaining three sets increasingly squeezed the range of transformed projection values, using (integers in) the intervals $[0, 40)$, $[0, 30)$ and $[0, 20)$. This meant that, for the fourth set of runs, each spline was reduced to at most only 20 knots (even though quite likely 67 distinct values were present). The speed-up achieved by these approximations was noticeable, with runs from the fourth set finishing roughly four times faster than those from the first (which, by virtue of mapping to integers, was itself about five times quicker than using no approximation at all). The different approximations had no discernible effect on training or test error, so for the purpose of plotting, I have merged the results of runs from all four sets. This was a pleasing finding, as it suggested my crude approach to reducing the run time might have value.

For most choices of r, p_g and λ, the corresponding group of points is reasonably well clustered. This is particularly true for the cluster of light blue dots. These correspond to a very high value for the smoothing parameter ($\lambda = 1000$), which essentially enforces the linear model, and its prediction accuracy lies just above that of the lasso. By contrast, the very low value ($\lambda = 0.1$) has the greatest dispersion, indicating that for this level of flexibility, convergence was not sufficiently attained within the 3,000 iterations the method was run. Clustering of groups was also affected by the selection of r; the smaller value ($r = 0.5$) resulted in more defined clusters. Finally, the choice of prior probability seemed to have had limited effect.

The tightness of groups is indicative of the stability of the posterior probability estimates. However, even for the settings for which convergence was most an issue, the conclusions drawn

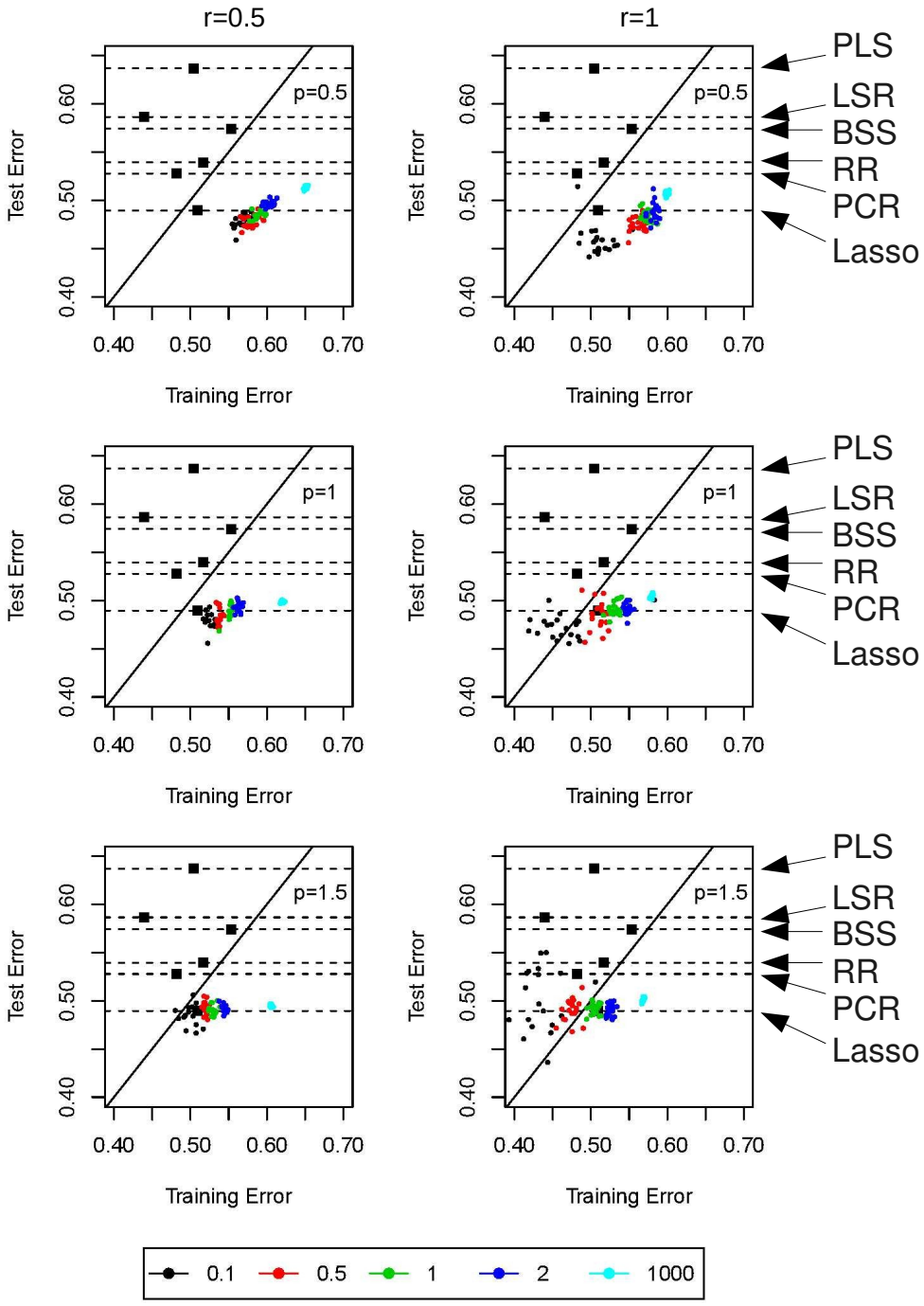


Figure 6.5: *Analysis of prostate cancer dataset. Each graph point plots the training error (x-axis) against test error (y-axis). In total, I considered 30 separate scenarios, running* Bayesian Projection Pursuit *20 times for each. The columns indicate choice of r; the left column denotes* $r = 0.5$*, the right column denotes* $r = 1$*. Each row considers a different value for the prior mean, as marked in the top right corner. Within each plot, I consider five different values for* λ*, ranging from 0.1 to 1000, designated by colour. For reference, the large black squares indicate the training and test error for the six methods considered in ESL. The relative performance of* Bayesian Projection Pursuit *can be assessed by counting under how many horizontal lines each point falls.*

from the data were generally the same. Using a posterior probability threshold of 0.5, *Bayesian Projection Pursuit* typically declared predictors 1 and 5 associated (`lcavol` and `svi`). Occasionally, the method also found evidence for predictor 2 (`lweight`). *Bayesian Projection Pursuit* obtained sparser results than five of the rival methods: LSR, RR, PCR and PLS automatically include all predictors in the final model, while the Lasso found evidence for five associations (predictors 1,2,4,5 and 8). The results of *Bayesian Projection Pursuit* were, in terms of sparsity, on a par with BSS, which declared only predictors 1 and 2 to be causal.

Presumably, the aim of this experiment was prognostic, asking how well `lpsa` could be predicted from the clinical factors. Therefore, it is satisfying that *Bayesian Projection Pursuit* has performed so well. Even though convergence was not perfect, consistently the method outperformed LSR, BSS, RR, PCR and PLS. Whether it outperformed the Lasso, depended on the choices of r and λ. The smoothing parameter is perhaps the hardest value to set. Ideally, as I discussed, λ would be treated as a variable and assigned its own prior. But, for the moment, this toy example would suggest that values close to 1 should give reasonable results, and provide an acceptable balance between overly cautious and overly zealous smoothing.

Instead of considering only each method's test prediction error, it is interesting to compare this with training prediction error. Averaged across a range of training and test set divides, ideally, we would like a method's point to lie on the diagonal line, indicating its training and test errors were equal. For five of the frequentist methods, the training error is lower than the test error, suggesting overfitting has occurred during parameter selection. Using this criterion, the best parameter choices for *Bayesian Projection Pursuit* for this problem appear to be $r = 0.5$ and $p = 1.5$, corresponding to the bottom left plot.

One set-back of *Bayesian Projection Pursuit*, if used for prediction, is obtaining a model which can be transferred. As the six methods featured in ESL are linear, their final models can readily be described by the regression coefficients. For *Bayesian Projection Pursuit*, as with *Sparse Partitioning*, no single model is returned. Instead, a distribution of possible partitions is found, this time each with its own set of direction coefficients. For the prostate cancer dataset, it was possible for me to test prediction accuracy because I had the test samples available, so could incorporate prediction of their response values at every iteration of the MCMC sampling. The only way to create a portable prediction model, would be to store the state of the Markov chain at each iteration, then reconstruct the run using a new set of test samples. While this is a straightforward operation for me to carry out, it would be fairly cumbersome for a clinician, who would much rather inspect the data visually with a simple classification rule. Unfortunately, this issue will be a hindrance for any nonlinear Bayesian regression method, and the only solution I can imagine is to create software as user friendly as possible.

Final Thoughts

Early on, I realised how easy it was to develop a successful regression method tailored to a particular scenario. For example, I could readily design a method searching only for multiplicative interactions, then demonstrate its superiority when such an interaction existed. The obvious drawback of such a method, is that its use will be heavily restricted, as it can only be applied to datasets which fall in line with its assumptions concerning the underlying relationship.

When designing *Sparse Partitioning*, I tried to take the opposite approach, and instead create a method as general as possible. The natural assumption is that generality comes at a cost; that a method will suffer reduced power as the model space becomes too large to feasibly search. Yet time and time again, I have been surprised that this has not seemed much of an issue. By contrast, I have observed that performance is greatly damaged when a method's assumptions are too restrictive for the dataset being analysed. I hope that with *Sparse Partitioning*, I offer a robust alternative to existing tools; my method seems to fare equally well under simple models, but comes into its own as the model becomes more complex.

Sparse Partitioning's drawback is the time it takes. Although I have applied the method to thousands of predictors, studies with hundreds of thousands, the ultimate goal if to be applicable to genome wide association studies, are still out of reach. With *Deterministic SP*, I have tried to address this issue, producing a method which completes in a fraction of the time, with what appears to be only a modest reduction in accuracy. Additionally, the deterministic version offers many attractive features for people put off by the uncertainty attached to Bayesian methods. Currently, there seem to be few, if any, methods able to consider multiple interaction models on a large scale, which suggests that there might be a market for such an approach.

Finally, in *Bayesian Projection Pursuit*, I have tried to take the idea of generality to the extreme, creating a method which can be applied to all types of predictors. So far, the early results appear promising, although once again, computation time becomes a major issue. Hopefully, I will be able to develop the algorithm so that it can be applied to very large datasets. But even if this does not prove possible, as it stands, I believe the method still represents a worthwhile addition to those currently available.

Software and Publication

All versions of *Sparse Partitioning* have been implemented and are currently available at the website of Simon Tavaré's Group on the University of Cambridge webpages. The current link is `http://www.compbio.group.cam.ac.uk/`. Alternatively, search for "Sparse Partitioning".

A manuscript detailing the methodology for the standard version of *Sparse Partitioning* has been accepted for publication by *The Annals of Applied Statistics*, to appear in 2011.

Bibliography

Akaike, H., 1974 A new look at the statistical model identification. Trans. Autom. Contr **19**: 716–723.

Albert, J., and S. Chib, 1993 Bayesian analysis of binary and polychotomous response data. J. Amer. Statist. Assoc. **88**: 669–679.

Albrechtsen, A., S. Castella, G. Andersen, T. Hansen, O. Pedersen, *et al.*, 2007 A Bayesian multilocus association method: allowing for higher-order interaction in association studies. Genetics **176**: 1197–1208.

Anderson, M., and C. Ter Braak, 2003 Permutation tests for multi-factorial analysis of variance. J. Stat. Comput. Sim. **73**: 85–113.

Aranzana, M., S. Kim, K. Zhao, E. Bakker, M. Horton, *et al.*, 2005 Genome-wide association mapping in *Arabidopsis* identifies previously known flowering time and pathogen resistance genes. PLoS Genet. **1**: e60.

Astle, W., and D. Balding, 2009 Population structure and cryptic relatedness in genetic association studies. Statist. Sci. **24**: 451–471.

Atwell, S., Y. Huang, B. Vilhjálmsson, G. Willems, M. Horton, *et al.*, 2010 Genome-wide association study of 107 phenotypes in *Arabidopsis thaliana* inbred lines. Nature **465**: 627–631.

Balding, D., 2006 A tutorial on statistical methods for population association studies. Nat. Rev. Genet. **7**: 781–791.

Benjamini, Y., and Y. Hochberg, 1995 Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Statist. Soc. B **57**: 289–300.

Biller, C., and L. Fahrmeir, 1997 Bayesian spline-type smoothing in generalized regression models. Computation. Stat. **12**: 131–151.

Breiman, L., 2004 Random forests. Machine Learning **45**: 5–32.

Breiman, L., J. Friedman, R. Olshen, and C. Stone, 1984 *Classification and regression trees*. Wadsworth Intern. Group.

Byrd, J., S. Jarvis, and A. Bhalerao, editors, 2008 *Reducing the Run-time of MCMC Programs by Multithreading on SMP Architectures*. IEEE International Symposium on Parallel and Distributed Processing.

Byrd, J., S. Jarvis, and A. Bhalerao, editors, 2010 *On the Parallelisation of MCMC by Speculative Chain Execution*. IEEE International Symposium on Parallel and Distributed Processing.

Carvalho, C., J. Chang, J. Lucas, J. Nevins, Q. Wang, *et al.*, 2008 High-dimensional sparse factor modeling: applications in gene expression genomics. J. Amer. Statist. Assoc. **103**: 1438–1456.

Chu, W., Z. Ghahramani, F. Falciani, and D. Wild, 2005 Biomarker discovery in microarray gene expression data with gaussian processes. Bioinformatics **21**: 3385–3393.

Clark, A., 1990 Inference of haplotypes from PCR-amplified samples of diploid populations. Mol. Biol. Evol. **7**: 111–122.

Conrad, D., M. Jakobsson, G. Coop, X. Wen, J. Wall, *et al.*, 2006 A worldwide survey of haplotype variation and linkage disequilibrium in the human genome. Nat. Genet. **38**: 1251–1260.

Cordell, H., 2009 Detecting gene-gene interactions that underlie human diseases. Nat. Rev. Genet. **10**: 392–404.

Curnow, R., and C. Dunnett, 1962 The numerical evaluation of certain multivariate normal integrals. Ann. Math. Statist. **33**: 571–579.

de Bruijn, N., 1958 *Asymptotic Methods in Analysis*. North-Holland Publishing Co.

Denison, D., and C. Holmes, 2003 Classification with Bayesian MARS. Machine Learning **50**: 159–173.

Dimas, A., 2009 *The role of regulatory variation in sculpting gene expression across human populations and cell types*. Ph.D. thesis, Darwin College, University of Cambridge.

Dobra, A., C. Hans, B. Jones, J. Nevins, G. Yao, *et al.*, 2004 Sparse graphical models for exploring gene expression data. J. Multivariate Anal. **90**: 196–212.

Eilers, P., and B. Marx, 1996 Flexible smoothing with B-splines and penalties. Statist. Sci. **11**: 89–102.

Engelhardt, B., and M. Stephens, 2010 Analysis of population structure: a unifying framework and novel methods based on sparse factor analysis. PLoS Genet. **6**: e1001117.

Fodor, I., 2002 A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Laboratory.

Fraser, A., editor, 2007 *Learning from the worm: predicting phenotype from genotype*.

Friedman, J., 1991 Multivariate adaptive regression splines. Ann. Statist. **19**: 1–67.

Friedman, J., and J. Tukey, 1974 A projection pursuit algorithm for exploratory data analysis. IEEE Trans. Comput. **C-23**: 881–889.

Galton, F., 1886 Regression towards mediocrity in hereditary stature. Journal of the Anthropological Institute **15**: 246–263.

Gelman, A., J. Carlin, H. Stern, and D. Rubin, 2004 *Bayesian Data Analysis*. Chapman & Hall/CRC.

Ghahramani, Z., 2010 Learning the structure of graphical models with latent variables. Cambridge Research Institute seminars.

Gilad, Y., S. Rifkin, and J. Pritchard, 2008 Revealing the architecture of gene regulation: the promise of eQTL studies. Trends Genet. **24**: 408–415.

Gilks, W., S. Richardson, and D. Spiegelhalter, 1996 *Markvo chain Monte Carlo in practice*. Chapman & Hall.

Goldstein, D., 2008 Common genetic variation and human traits. N. Engl. J. Med **360**: 1696–1698.

Golub, G., and C. Reinsch, 1970 Singular value decomposition and least squares solutions. Numer. Math. **14**: 403–420.

Hahn, L., M. Ritchie, and J. Moore, 2002 Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. Bioinformatics **19**: 376–382.

Hans, C., A. Dobra, and M. West, 2007 Shotgun Stochastic Search for "large p" regression. J. Amer. Statist. Assoc. **102**: 507–516.

Hardy, G., 1908 Mendelian proportions in a mixed population. Science **28**: 49–50.

Hastie, T., and R. Tibshirani, 1990 *Generalized additive models*. Chapman & Hall/CRC.

Hastie, T., R. Tibshirani, and J. Friedman, 2001 *The Elements of Statistical Learning*. Springer.

Hastings, W., 1970 Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**: 97–109.

Hein, J., M. Schierup, and C. Wiuf, 2005 *Gene Genealogies, Variation and Evolution*. Oxford University Press.

Hoeting, J., D. Madigan, A. Raftery, and C. Volinsky, 1999 Bayesian model averaging: a tutorial. Stat. Sci. **14**: 382–401.

Hoggart, C., J. Whittaker, M. De Iorio, and D. Balding, 2008 Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies. PLoS Genet. **4**: e10000130.

Howie, B., P. Donnelly, and J. Marchini, 2009 A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet. **5**: e1000529.

Hudson, R., 2002 Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics **18**: 337–338.

Jeffreys, H., 1935 Some tests of significance, treated by the theory of probability. Math. Proc. Cambridge **31**: 203–222.

Johanson, U., J. West, C. Lister, S. Michaels, R. Amasino, *et al.*, 2000 Molecular analysis of FRIGIDA, a major determinant of natural variation in *Arabidopsis* flowering time. Science **290**: 344–347.

Kang, H., J. Sul, S. Service, N. Zaitlen, S. Kong, *et al.*, 2010 Variance component model to account for sample structure in genome-wide association studies. Nat. Genet. **42**: 348–354.

Knijnenburg, T., L. Wessels, and M. Reinders, 2009 Fewer permutations, more accurate p-values. Bioinformatics **25**: i161–i168.

Kooperberg, C., and I. Ruczinski, 2005 Identifying interacting SNPs using Monte Carlo logic regression. Genet. Epidemiol. **28**: 157–170.

Kruglyak, L., 1999 Prospects for whole-genome linkage disequilibrium mapping of common disease genes. Nat. Genet. **22**: 139–144.

Kuo, L., and B. Mallick, 1998 Variable selection for regression models. Sankhyā Ser. B **60**: 65–81.

Maher, B., 2008 Personal genomes: the case of the missing heritability. Nature **456**: 18–21.

Mailund, T., S. Besenbacher, and M. Schierup, 2006 Whole genome association mapping by incompatibilities and local perfect phylogenies. Bioinfomatics **7**: 454.

Maini, M., R. Gilson, N. Chavda, S. Gill, A. Fakoya, *et al.*, 1996 Reference ranges and sources of variability of CD4 counts in HIV-seronegative women and men. Genitourin. Med. **72**: 27–31.

Manolio, T., L. Brooks, and F. Collins, 2008 A HapMap harvest of insights into the genetics of common disease. J. Clin. Invest. **118**: 1590–1605.

Manolio, T., F. Collins, N. Cox, D. Goldstein, L. Hindorff, *et al.*, 2009 Finding the missing heritability of complex diseases. Nature **461**: 747–753.

Marchini, J., P. Donnelly, and L. Cardon, 2005 Genome-wide strategies for detecting multiple loci that influence complex diseases. Nat. Genet. **37**: 413–417.

Marchini, J., and B. Howie, 2010 Genotype imputation for genome-wide association studies. Nat. Rev. Genet. **11**: 499–511.

Marchini, J., B. Howie, S. Myers, G. McVean, and P. Donnelly, 2007 A new multipoint method for genome-wide association studies by imputation of genotypes. Nat. Genet. **39**: 906–913.

McCarthy, M., G. Abecasis, L. Cardon, D. Goldstein, J. Little, *et al.*, 2008 Genome-wide association studies for complex traits: consensus, uncertainty and challenges. Nat. Rev. Genet. **10**: 356–369.

Mendel, G., 1865 Versuche uber pflanzen-hybriden. Verhandlungen des naturforschenden Vereines in Brünn **4**: 3–47.

Mukherjee, S., S. Pelech, R. Neve, W. Kuo, S. Ziyad, *et al.*, 2009 Sparse combinatorial inference with an application in cancer biology. Bioinformatics **25**: 265–271.

Murray, I., and Z. Ghahramani, 2005 A note on the evidence and Bayesian Occam's razor. Technical report, Gatsby Unit.

Nelson, M., S. Kardia, R. Ferrell, and C. Sing, 2001 A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation. Genome Res. **11**: 458–470.

Nordborg, M., T. Hu, Y. Ishino, J. Jhaveri, C. Toomajian, *et al.*, 2005 The pattern of polymorphism in *Arabidopsis thaliana*. PLoS Genet. **3**: e196.

Nordborg, M., and S. Tavaré, 2002 Linkage disequilibrium: what history has to tell us. Trends Genet. **18**: 83–90.

Park, M., and T. Hastie, 2008 Penalized logistic regression for detecting gene interactions. Biostatistics **9**: 30–50.

Patterson, N., A. Price, and D. Reich, 2006 Population structure and eigenanalysis. PLoS Genet. **2**: e190.

Pollock, D., 1999 *A handbook of time-series analysis, signal processing and dynamics. Chapter 11: Smoothing with Cubic Splines*. Academic Press, 307–310.

Prentice, R., and R. Pyke, 1979 Logistic disease incidence models and case-control studies. Biometrika **66**: 403–411.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery, 2007 *Numerical Recipes. Chapter 2: Solution of Linear Algebraic Equations*. Cambridge University Press, 81–82.

Price, A., N. Patterson, R. Plenge, M. Weinblatt, N. Shadick, *et al.*, 2006 Principal components analysis corrects for stratification in genome-wide association studies. Nat. Genet. **38**: 904–909.

Pritchard, J., and M. Przeworski, 2001 Linkage disequilibrium in humans: models and data. Am. J. Hum. Genet **69**: 1–14.

Pritchard, J., M. Stephens, and P. Donnelly, 2000 Inference of population structure using multilocus genotype data. Genetics **155**: 945–959.

Psychiatric GWAS Consortium Coordinating Committee, 2009 Genomewide association studies: History, rationale, and prospects for psychiatric disorders. Am. J. Psychiatry **166**: 540–556.

R Development Core Team, 2008 *R: A language and environment for statistical computing*. ISBN 3-900051-07-0.

Ramsay, J., and B. Silverman, 2006 *Functional Data Analysis*. Springer.

Ravikumar, P., 2009 Sparse additive models. J. R. Statist. Soc. B **71**: 1009–103.

Reich, D., and E. Lander, 2001 On the allelic spectrum of human disease. Trends Genet. **17**: 502–510.

Reinsch, C., 1967 Smoothing by spline functions. Numer. Math. **10**: 177–183.

Risch, N., and K. Merikangas, 1996 The future of genetic studies of complex human diseases. Science **273**: 1516–1517.

Ruczinski, I., C. Kooperberg, and M. LeBlanc, 2003 Logic regression. J. Comput. Graph. Statist. **12**: 475–511.

Scheet, P., and M. Stephens, 2006 A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. Am. J. Hum. Genet. **78**: 629–644.

Seaman, S., and S. Richardson, 2004 Equivalence of prospective and retrospective models in the Bayesian analysis of case-control studies. Biometrika **91**: 15–25.

Shao, H., L. Burrage, D. Sinasac, A. Hill, S. Ernest, *et al.*, 2008 Genetic architecture of complex traits: large phenotypic effects and pervasive epistasis. PNAS **105**: 19910–19914.

Shindo, C., M. Aranzana, C. Lister, C. Baxter, C. Nicholls, *et al.*, 2005 Role of FRIGIDA and FLOWERING LOCUS C in determining variation in flowering time of *Arabidopsis thaliana*. Plant Physiol. **138**: 1163–1173.

Šidák, Z., 1967 Rectangular confidence regions for the means of multivariate normal distributions. J. Amer. Statist. Assoc. **62**: 626–63347.

Solberg, L., W. Valdar, D. Gauguier, G. Nunez, A. Taylor, *et al.*, 2006 A protocol for high-throughput phenotyping, suitable for quantitative trait analysis in mice. Mamm. Genome **17**: 129–146.

Speed, D., editor, 2008 *Parallelising MCMC in Sparsity Problems*. SciComp@Cam minitalks.

Stamey, T., J. Kabalin, J. McNeal, I. Johnstone, F. Freiha, *et al.*, 1989 Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients. J. Urology **16**: 1076–1083.

Stephens, M., editor, 2010 *Association analysis for multivariate outcomes*. Bloomsbury Centre for Genetic Epidemiology and Statistics Seminar Series.

Stephens, M., N. Smith, and P. Donnelly, 2001 A new statistical method for haplotype reconstruction from population data. Am. J. Hum. Genet. **68**: 978–989.

Stranger, B., M. Forrest, M. Dunning, C. Ingle, C. Beazley, *et al.*, 2007 Relative impact of nucleotide and copy number variation on gene expression phenotypes. Science **315**: 848–853.

The International HapMap Consortium, 2003 The International HapMap Project. Nature **426**: 789–796.

The International HapMap Consortium, 2004 A haplotype map of the human genome. Nature **437**: 1299–1320.

The International HapMap Consortium, 2007 A second generation human haplotype map of over 3.1 million SNPs. Nature **449**: 851–861.

The Wellcome Trust Case Control Consortium, 2007 Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature **447**: 661–678.

The Wellcome Trust Case Control Consortium, 2010 Genome-wide association study of CNVs in 16,000 cases of eight common diseases and 3,000 shared controls. Nature **464**: 713–720.

Thomas, D., 2010 Gene-environment-wide association studies: emerging approaches. Nat. Rev. Genet. **11**: 259–272.

Tibshirani, R., 1996 Regression shrinkage and selection via the Lasso. J. R. Statist. Soc. B **58**: 267–288.

Verzilli, J., N. Stallard, and J. Whittaker, 2006 Bayesian graphical models for genomewide association studies. Am. J. Hum. Genet. **79**: 100–112.

Visscher, P., 2008 Sizing up human height variation. Nat. Genet. **40**: 489–490.

W de Bakker, P., R. Yelensky, I. Pe'er, S. Gabriel, M. Daly, *et al.*, 2005 Efficiency and power in genetic association studies. Nat. Genet. **37**: 1217–1223.

Wahba, G., 1978 Improper priors, spline smoothing and the problem of guarding against model errors in regression. J. R. Statist. Soc. B **40**: 364–372.

Wakefield, J., 2009 Bayes factors for genome-wide association studies: comparison with p-values. Genet. Epidemiol. **33**: 79–86.

Wang, H., Y. Zhang, X. Li, G. Masinde, S. Mohan, *et al.*, 2005 Bayesian shrinkage estimation of quantitative trait loci parameters. Genetics **170**: 465–480.

Wassserman, L., 2000 Bayesian model selection and model averaging. J. Math. Psychol. **44**: 92–107.

Weinberg, W., 1908 Über den Nachweis der Vererbung beim Menschen. Verein Vaterländ Naturk, Wurtemberg Jahresh **64**: 368–382.

Werft, W., and A. Benner, 2010 glmperm: a permutation of regressor residuals test for inference in generalized linear models. The R Journal **2**: 39–43.

Wright, S., 1922 Coefficients of inbreeding and relationship. Amer. Nat. **61**: 330–338.

Yi, N., and S. Xu, 2008 Bayesian Lasso for quantitative trait loci mapping. Genetics **179**: 1045–1055.

Ypma, T., 1995 Historical development of the Newton-Raphson method. SIAM Rev. **37**: 531–551.

Yu, J., G. Pressoir, W. Briggs, I. Vroh Bi, M. Yamasaki, *et al.*, 2006 A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. Nat. Genet. **38**: 203–208.

Zhang, M., K. Montooth, M. Wells, A. Clark, and D. Zhang, 2005 Mapping multiple quantitative trait loci by Bayesian classification. Genetics **169**: 2305–2318.

Zhang, W., J. Zhu, E. Schadt, and J. Liu, 2010 A Bayesian partition method for detecting pleiotrophic and epistatic eQTL modules. PLoS Comp. Bio. **6**: e1000642.

Zhang, Y., B. Jiang, J. Zhu, and J. Liu, 2011 Bayesian methods for detecting epistatic interactions from genetic data. Ann. Hum. Genet. **75**: 183–193.

Zhang, Y., and J. Liu, 2007 Bayesian inference of epistatic interactions in case-control studies. Nat. Genet. **39**: 1167–1173.

Zhang, Y., and S. Xu, 2005 A penalized maximum likelihood method for estimating epistatic effects of QTL. Heredity **95**: 96–104.

Zhao, K., M. Aranzana, S. Kim, C. Lister, C. Shindo, *et al.*, 2007 An *Arabidopsis* example of association mapping in structured samples. PLoS Genet. **3**: e4.