Inferential stability in systems biology

by

Paul Kirk

A thesis submitted for the degree of
Doctor of Philosophy of Imperial College London

Division of Molecular Biosciences
Imperial College London
London SW7 2AZ, England

Typeset in Times by LaTeX

This report is the result of my own work.

No part of this dissertation has already been, or is currently being submitted by the author for any other degree or diploma or other qualification.

This dissertation does not exceed 100,000 words, excluding appendices, bibliography, footnotes, tables and equations. It does not contain more than 150 figures.

This work is supported by a Wellcome Trust grant and completed in the Division of Molecular Biosciences at Imperial College, London.

Abstract

The modern biological sciences are fraught with statistical difficulties. Biomolecular stochasticity, experimental noise, and the "large p, small n" problem all contribute to the challenge of data analysis. Nevertheless, we routinely seek to draw robust, meaningful conclusions from observations. In this thesis, we explore methods for assessing the effects of data variability upon downstream inference, in an attempt to quantify and promote the stability of the inferences we make.

We start with a review of existing methods for addressing this problem, focusing upon the bootstrap and similar methods. The key requirement for all such approaches is a statistical model that approximates the data generating process.

We move on to consider biomarker discovery problems. We present a novel algorithm for proposing putative biomarkers on the strength of both their predictive ability and the stability with which they are selected. In a simulation study, we find our approach to perform favourably in comparison to strategies that select on the basis of predictive performance alone.

We then consider the real problem of identifying protein peak biomarkers for HAM/TSP, an inflammatory condition of the central nervous system caused by HTLV-1 infection. We apply our algorithm to a set of SELDI mass spectral data, and identify a number of putative biomarkers. Additional experimental work, together with known results from the literature, provides corroborating evidence for the validity of these putative biomarkers.

Having focused on static observations, we then make the natural progression to time course data sets. We propose a (Bayesian) bootstrap approach for such data, and then apply our method in the context of gene network inference and the estimation of parameters in ordinary differential equation models. We find that the inferred gene networks are relatively unstable, and demonstrate the importance of finding distributions of ODE parameter estimates, rather than single point estimates.

Contents

Acknowledgements

I would like to thank my supervisors Michael Stumpf and Sylvia Richardson for providing guidance, advice, and support. I also thank Alex Lewin for her time, suggestions and many interesting discussions.

I thank all of the members of the Theoretical Systems Biology group for providing a helpful and friendly environment. In particular: Chris Barnes, Georgia Chan, Kamil Erguler, Nathan Harmston, Maxime Huvet, Liam Kelly, Dan Silk and Tina Toni.

I am grateful for the funding and support from the Wellcome Trust, and for the opportunities afforded to me by the 4-year doctoral programme at Imperial College.

The HTLV-1 project discussed in Chapter 3 was devised by Charles Bangham. Aviva Witkover performed the SELDI experiments; Alan Courtney conducted the protein identifications; and Graham Taylor recruited patients. Special thanks must go to Charles and Aviva, who have been very patient throughout the iterative process of experimental and theoretical work.

The content of Appendix B arose as a result of discussions with Matt Nunes, Liam Kelly, and David Balding. I hope that we might be able to take this work forward at some point in the future.

This dissertation was greatly improved by those who commented on and proof read it; thanks go to Michael, Sylvia and Liam.

I thank the friends who have provided moral support (and occasional necessary diversions) at various points during my PhD. Thanks to $\Theta\Sigma B$ Hyde Park Relay Teams α and $\aleph$, the Bioinformatics badminton group, and my training partner/squash opponent for sport relief.

Finally, I thank my family for their love and support.

Chapter 1

Introduction

Abstract *This thesis is broadly concerned with the approximation of data-generating processes (DGPs) in systems biology. Our principal aim is to assess the stability of inferences and conclusions drawn from biological and biomedical data. In this chapter, we describe our motivations, discuss existing methods for the approximation of DGPs, and explain how inferential stability may be assessed.*

Outline *In Sections 1.1 and 1.2, we motivate our work and summarise the key statistical challenges routinely encountered in bioinformatics and systems biology. As a first step toward drawing meaningful conclusions from our data, it is vital to quantify the effects of these difficulties. We are here interested in determining whether or not our inferences are robust to realistic perturbations of the data. We therefore explain in Section 1.3 how approximations of DGPs may be used to assess inferential stability. We consider a number of methods, focussing on bootstrap and subsampling techniques. We discuss "Bayesian bootstrap" approaches and make a straightforward connection between these and the posterior predictive checking framework. We finish in Section 1.6 with an overview of the thesis.*

1.1 Motivation

Concerns about the reproducibility of results in systems biology abound, both in the scientific literature (Baggerly *et al.*, 2004; Ein-Dor *et al.*, 2006; Zhang *et al.*, 2008, 2009) and the popular press (Pollack, 2004). Since the outcomes of systems biology research may have implications for future medical treatments and practices, reproducibility is clearly vital. However, given the many statistical challenges associated with systems biology data (see Section 1.2), it can often be difficult to determine whether our conclusions arise as a result of the underlying biology, or if they are a side-effect of biases and/or noise

in the experimental procedure (Marshall, 2004). In the latter case, reproducibility will almost inevitably be affected adversely.

In order to assess reproducibility, we would ideally repeat experiments many times, and see if we consistently draw the same (or similar) inferences. In practice, we are limited by the costs and resources required to conduct experimental studies. We therefore seek alternative, statistical approaches that approximate the experimental *data generating process* (DGP). In this way we assess the *stability* of our inferences; i.e. the degree to which they vary in light of realistic perturbations to the data. Ultimately, we hope that this will help to improve the reliability of reported conclusions in systems biology and the biomedical sciences more broadly.

1.2 Statistical challenges in systems biology

The past decade has seen a dramatic increase in the amount — and the variety — of biological data. This has been driven by the advent of modern high-throughput technologies that enable measurements to be taken on large numbers of biological and biomolecular entities (such as genes or proteins) rapidly and at low cost. One obvious and very important example of this is the microarray, which routinely allows the expression levels of thousands of genes to be measured simultaneously. The microarray and similar technologies have revolutionised the modern biomedical and life sciences, providing a means by which to probe the complex systems that underpin the functions performed by cells and organisms.

These experimental advances — and the resulting quantities of data they generate — have given rise to novel challenges for statisticians and data analysts. We here provide a brief overview of some of the most fundamental, all of which have an impact upon the work presented in this thesis. For the sake of brevity and focus, we do not cover all of these challenges in depth; further information may be found in the references herein.

1.2.1 The "large p, small n" problem

The number of covariates, p, on which we have measurements is usually many more than the number of available observations, n. This leads to characteristically "wide" data matrices when analysing microarray (Efron *et al.*, 2001), proteomics (Barla *et al.*, 2008), and other 'omics data sets (Broadhurst and Kell, 2006). Within this "large p, small n" paradigm (West, 2003), classical statistical techniques generally fail (Ochs, 2010). Moreover, it becomes particularly important to address challenges such as overfitting and multiple hypothesis testing (see Section 1.2.2). A host of different approaches have been proposed to mitigate and overcome the challenges presented when working within this paradigm. For example, a variety of dimension reduction and feature selection techniques

are routinely employed in order to decrease the value of p by (for example) projecting into a lower dimensional space, compressing information, or removing redundant covariates (see Saeys *et al.*, 2007, for a review). At the same time, James-Stein shrinkage (Opgen-Rhein and Strimmer, 2007a), regularisation (Kim and Park, 2004), and Bayesian (Fox and Dimmic, 2006) versions of classical techniques such as the t-test have been considered in order to extend their applicability.

1.2.2 The multiple comparisons problem

Classical frequentist hypothesis tests seek to assess the significance of an observed statistic, assuming some (parametric or nonparametric) null model for that statistic (Fisher, 1925). For example, in *differential expression* analyses, we are interested in determining whether or not the difference in the mean expression level of a gene in two different conditions is significant, and might assume Student's t-distribution for the null model. If the observed value of the statistic is in the tails of the null distribution (as quantified by a p-value), then we might decide to reject the null hypothesis. The critical p-value below which the null hypothesis is rejected is known as the *significance level*, α. In scientific studies, $\alpha = 0.05$ is a particular common choice, which corresponds to a 5% chance of incorrectly rejecting the null hypothesis (i.e. a 5% chance of a false positive/Type I error). The difficulty with this approach is that, when we perform a very large number of tests (which, for simplicity, are usually assumed to be independent), our control over the probability of falsely rejecting the null hypothesis for at least one may become quite weak. One way in which this may be formalised is by considering the *familywise error rate* (FWER), which represents the experiment-wide significance level and may be interpreted as the probability of incorrectly rejecting the null hypothesis in at least one of the individual tests. The FWER is defined by,

$$\alpha_{\text{FWER}} = 1 - (1 - \alpha)^p, \tag{1.1}$$

where p is the total number of tests (assumed to be independent), and α is the significance level for each individual test. So, if $\alpha = 0.05$ and we were performing $p = 100$ tests, then $\alpha_{\text{FWER}} \approx 0.99$.

A number of procedures to correct for multiple comparisons by controlling the FWER have been proposed, including the Bonferonni, Šidàk (Šidàk, 1968) and Holm-Bonferonni (Holm, 1979) methods. We refer to Shaffer (1995) for a comprehensive review of these and several other approaches. An alternative methodology is the Benjamini-Hochberg procedure (Benjamini and Hochberg, 1995), which controls the *false discovery rate* (FDR). The FDR is simply the expected Type I error rate (i.e. the expected proportion of incorrectly rejected hypotheses amongst all rejected hypotheses).

1.2.3 Covariate interdependencies

In the previous section, we briefly touched upon the common assumption of independence when performing multiple statistical comparisons. Of course, in biological systems, strong dependencies between the various biomolecular players mean that this assumption must generally be regarded as a gross simplification. An obvious but important example of this is the presence of dependencies amongst and between mRNA and protein expression levels, which arise as a result of gene regulation mechanisms. In the context of differential expression analyses, several variants of the t-test have been proposed to take these into account (Tibshirani and Wasserman, 2006; Lai, 2008; Zuber and Strimmer, 2009).

Covariate dependencies are not mere statistical nuisances, however. Rather, they represent the observable effects of intricate networks of interactions between different biomolecular species, and hence provide an opportunity to elucidate this complex behaviour. Statistical descriptions of these interactions are usually considered in the context of *network inference* (Butte *et al.*, 2000; Margolin *et al.*, 2006; Schäfer *et al.*, 2006; Lèbre, 2009). As well as empirical representations, it is also increasingly common to consider mechanistic models (such as ordinary, stochastic and partial differential equations). These models are typically parametric, and hence parameter inference is a key challenge for model fitting. This task is complicated not only by the vast numbers of parameters that must often be inferred (for example, Schoeberl *et al.*, 2002, propose a model of a cell signalling pathway comprising 94 state variables and 95 parameters), but also by inherent properties of the models, such as parameter non-identifiability (Gutenkunst *et al.*, 2007) and the presence of bifurcations (Kirk *et al.*, 2008). Overall, regardless of whether we consider empirical or mechanistic models, adequately capturing the complex interdependencies of the underlying biological system remains an ongoing challenge.

1.2.4 Sources of stochasticity

Exacerbating the problems of small sample sizes, large numbers of covariates, and complex dependency structures, is the ubiquity of stochasticity. At the experimental level, the high throughput technologies that generate the vast majority of data sets that we study have often been criticised for being highly noisy (Marshall, 2004). Although more recent assessments tend to be more optimistic (Klebanov and Yakovlev, 2007), the principal difficulty remains that the number of technical replicates that may be obtained is seriously limited by the expenses associated with data generation (Lee *et al.*, 2000). Given time, we may reasonably expect that these technologies will become more precise and that data will become cheaper to generate. However, even assuming that experimental sources of noise may be eliminated, biomolecular systems are implicitly stochastic, and — given the relatively small numbers of molecules that may be involved in a given process — deterministic approximations will often be inadequate (Wilkinson, 2006; Phillips *et al.*, 2009). Moreover, we are often concerned with *populations* of cells or organisms, and hence must be aware of the variability between different individuals.

The main practical implication of these various sources of stochasticity is that, when we repeat experiments, we expect to obtain (slightly) different results each time. We must therefore be careful not to overfit to any one particular data set, or to draw conclusions that are not robust to the variability in our observations.

1.3 Assessing inferential stability

Despite the statistical challenges described in Section 1.2, we seek to draw general conclusions and make useful inferences about the nature of the biological systems being studied. Perhaps the most important quality that we desire our conclusions to have is for them to be *reproducible*, and not overly specific to the particular (small) set of measurements that was originally observed. From a statistical inference perspective, we wish to avoid overfitting and to ensure that our inferences generalise well to new, previously unseen data sets.

In this thesis, we shall be concerned with the estimation of structures (models) and quantities (typically model parameters) from biological data. One of our recurring aims will be to assess the *stability* of these estimates. By *stability* we mean the robustness of our estimates to realistic perturbations of the data. The way in which we shall assess stability is by approximating the underlying data generating process (DGP) using empirical, data-driven approaches. Informally, our aim is to determine how the variability in the observed data translates into variability in our conclusions.

Before introducing specific methodologies for approximating DGPs, we consider a framework for assessing inferential stability using the "plug-in principle" (Efron and Tibshirani, 1993).

1.3.1 The plug-in principle

We denote the true, underlying DGP by F, and assume that we have an observed data set $D^{\text{obs}} = \{\mathbf{x}_1^{\text{obs}}, \ldots, \mathbf{x}_n^{\text{obs}}\}$, so that $\mathbf{x}_1^{\text{obs}}, \ldots, \mathbf{x}_n^{\text{obs}}$ represents a random sample of size n drawn from F. Following Efron (1979), we construct the sample probability distribution, $\hat{F}$, which puts mass $1/n$ at each sample point. This probability distribution is therefore defined by,

$$p(\mathbf{x}) = \frac{\text{Multiplicity of } \mathbf{x} \text{ in } D^{\text{obs}}}{n}, \tag{1.2}$$

where the "multiplicity of $\mathbf{x}$ in D^{obs}" is simply the number of times $\mathbf{x}$ appears in D^{obs}. The *plug-in principle* is the practice of estimating aspects of F by the corresponding aspects of $\hat{F}$ (Efron and Tibshirani, 1993).

More precisely, suppose $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a random sample of size n from the distribution F, and let $s = T(D)$ be a summary statistic, where T is some deterministic

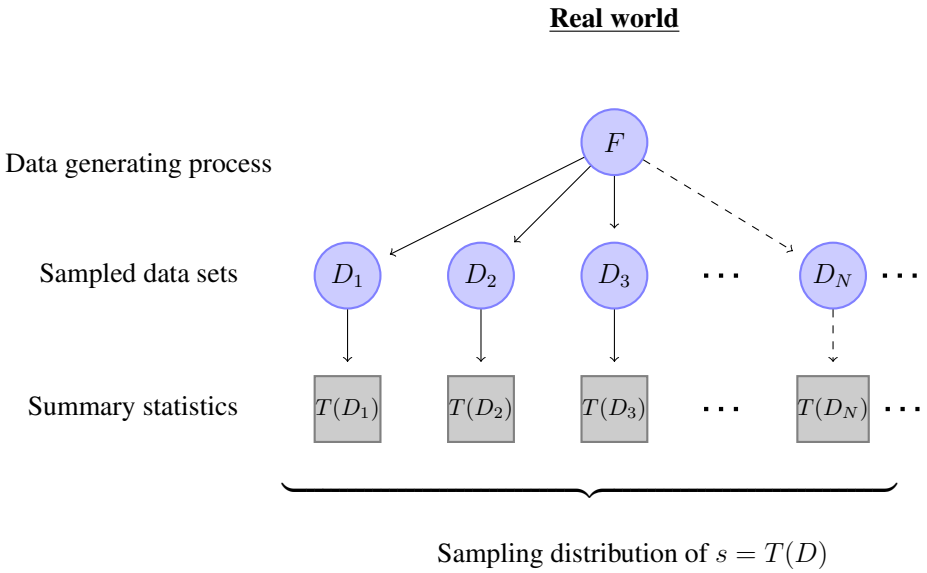
function of the elements of D. For example, we might have $T(D) = \sum_{i=1}^{n} \mathbf{x}_i/n$, in which case s is simply the sample mean. In the frequentist tradition, we quantify the variability in the summary statistic by considering its *sampling distribution*. This is the (hypothetical) distribution of s that we would obtain if we were to repeatedly sample sets D_i of size n from F, and were to calculate $T(D_i)$ for each one. In practice, we might not be interested in the whole sampling distribution, but may be content with summaries, such as the expectation or standard error of s. The real difficulty is that we cannot usually determine the sampling distribution analytically (unless strong parametric assumptions are made), and brute force alternatives (sampling exhaustively from F) are usually prohibitively expensive.

The plug-in principle tells us to consider $\hat{F}$ in the place of F. This is useful, since we may obtain random samples of size n from $\hat{F}$ computationally: we simply draw n observations with replacement from D^{obs} (as we discuss in Section 1.4.1.1, this defines the *nonparametric bootstrap*). If we repeat this many times, we generate a large number, B, of *replicate data sets* that we denote $D^{\text{rep}}_{(1)}, \ldots, D^{\text{rep}}_{(B)}$. We then calculate the value of the summary statistic for each replicate data set , $s^{\text{rep}}_{(i)} = T(D^{\text{rep}}_{(i)})$, for $i = 1, \ldots, B$. The resulting frequency distribution of s^{rep} (which is sometimes termed the *bootstrap distribution* of the summary statistic) is used in place of the sampling distribution of s. We may also use the replicate data sets in order to obtain Monte Carlo estimates of (for example) the expectation and standard error of s^{rep} with respect to the distribution $\hat{F}$ (see Metropolis and Ulam, 1949; Robert and Casella, 2004, for details of Monte Carlo methods). As $B \to \infty$, these estimates approach the true expectation and standard error (with respect to $\hat{F}$). Again, these estimates are used as surrogates for the quantities that we would ideally derive from the true, unknown distribution, F.

Although the original formulation of the plug-in principle assumes $\hat{F}$ to be the sample probability distribution function of Equation 1.2, the *parametric bootstrap* (Section 1.4.1.2) extends the principle to more general approximations of the DGP. The idea is always the same, however: we simulate a large number of data sets from the approximate DGP, and then use these to obtain the bootstrap distribution of the statistic of interest. A schematic view of this procedure is provided in Figure 1.1 (an elaboration upon Figure 1 from Efron, 2003), which draws a comparison between the "real world" defined by F, and the so-called "bootstrap world" (Efron and Tibshirani, 1993) defined by $\hat{F}$.

1.4 Methods for approximating data generating processes

The plug-in principle provides us with a framework for assessing the stability of a summary statistic, s, or — more generally — of any quantity estimated from the data. The lower the variability in the bootstrap distribution of s^{rep}, the more stable we will believe s to be. The main requirement needed to apply the plug-in principle is an approximation, $\hat{F}$, of the DGP, from which we may draw replicated data sets. We here consider a number of approaches for approximating the DGP, focusing in particular upon bootstrap and



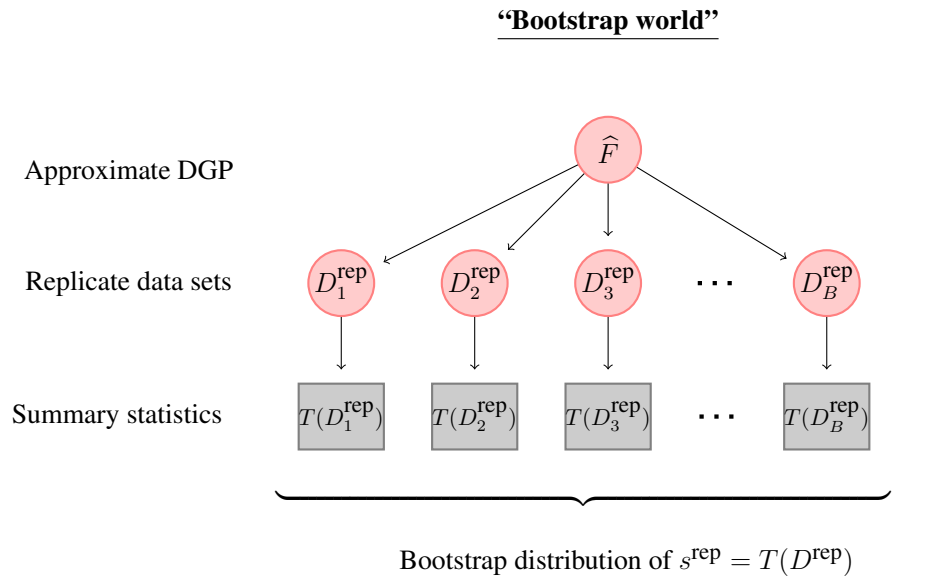


Figure 1.1: Illustration of the plug-in principle and comparison between the real world and the "bootstrap world". We simply use the approximate DGP, $\hat{F}$, wherever we would ideally use the true, unknown DGP, F. We simulate a large number of times from the approximate DGP and calculate the statistic of interest for each replicate data set. We hence obtain the bootstrap distribution for the summary statistic, which we use in the place of the unknown sampling distribution.

subsampling methods. We discuss the Bayesian bootstrap of Rubin (1981b), and draw comparisons with the posterior predictive checking framework of Gelman *et al.* (1996).

1.4.1 The Bootstrap

As we have already touched upon, the bootstrap is a well-known and widely applied method for assessing properties of an inferred quantity or statistical estimator (Efron, 1979; Efron and Tibshirani, 1993). There have been many applications of the bootstrap to biological problems. Amongst the earliest of these is the work of Felsenstein (1985) (later updated by Efron *et al.*, 1996), who used a bootstrapping procedure to assign confidence intervals to phylogenies. Other examples include (to name but a few): assessing the reliability of conclusions drawn from clustering expression data (Kerr and Churchill, 2001); constructing "robust" estimates of gene networks (Imoto *et al.*, 2004); and assigning confidence scores to protein-protein interactions (Friedel *et al.*, 2009).

Bootstrap approaches fall into two broad categories: parametric and nonparametric. The nonparametric bootstrap (Efron, 1979) is the more widely applied, and is commonly referred to as *the* bootstrap. We consider both nonparametric and parametric varieties in Sections 1.4.1.1 and 1.4.1.2.

1.4.1.1 Nonparametric bootstrap

In the nonparametric case, we obtain bootstrap data sets by drawing samples of size n *with replacement* from the original data set. For example, if $n = 6$, then the following are possible bootstrap data sets derived from D^{obs},

$$\text{NPBS1:} \quad \{\mathbf{x}_6^{\text{obs}}, \quad \mathbf{x}_3^{\text{obs}}, \quad \mathbf{x}_1^{\text{obs}}, \quad \mathbf{x}_1^{\text{obs}}, \quad \mathbf{x}_2^{\text{obs}}, \quad \mathbf{x}_5^{\text{obs}}\} \tag{1.3}$$

$$\text{NPBS2:} \quad \{\mathbf{x}_3^{\text{obs}}, \quad \mathbf{x}_6^{\text{obs}}, \quad \mathbf{x}_1^{\text{obs}}, \quad \mathbf{x}_3^{\text{obs}}, \quad \mathbf{x}_2^{\text{obs}}, \quad \mathbf{x}_2^{\text{obs}}\} \tag{1.4}$$

$$\text{NPBS3:} \quad \{\mathbf{x}_5^{\text{obs}}, \quad \mathbf{x}_1^{\text{obs}}, \quad \mathbf{x}_3^{\text{obs}}, \quad \mathbf{x}_1^{\text{obs}}, \quad \mathbf{x}_4^{\text{obs}}, \quad \mathbf{x}_1^{\text{obs}}\} \tag{1.5}$$

Basic properties of the nonparametric bootstrap Note that, as a result of sampling with replacement, our bootstrap data sets can contain the same observation more than once (and thus are strictly multisets, although we shall suppress this distinction throughout). Since each such set is of fixed size n and contains only elements from D, there is a finite number, n_{BS}, of distinct bootstrap samples that may be obtained; namely,

$$n_{BS} = \binom{2n-1}{n}. \tag{1.6}$$

In practice, however, we rarely generate all possible bootstrap data sets, since n_{BS} is large even for relatively small n (for example, when $n = 10$, there are already 92,378 possible bootstrap samples, and for $n = 20$ there are nearly 7×10^{10}).

One further property of the nonparametric bootstrap that we briefly mention concerns the average number of observations in D that are *left out* of each replicate data set. The

probability of a particular observation $\mathbf{x}_i$ not appearing in our bootstrap data set is given by $(1 - \frac{1}{n})^n$. As n grows large, this probability tends to $\exp(-1) \approx 0.368$, and hence the average number of observations left out of each bootstrap data set is approximately $0.368n$. It follows that the average number of distinct observations in each bootstrap sample is $0.632n$. This realisation is the motivation behind the *.632 bootstrap estimator* of prediction error, for further details of which we refer to Efron and Tibshirani (1993, 1997).

Connection to the multinomial distribution Although usually described algorithmically in terms of how bootstrap data sets are generated, there is an implicit probability model behind the nonparametric bootstrap. We first note that — given D^{obs} — any nonparametric bootstrap data set, D^{rep}, is completely described by the multiplicities with which each of the elements of D^{obs} appears in D^{rep}. For example, NPBS1 (Equation 1.3) is completely described by the vector $\mathbf{N}_1 = [2, 1, 1, 0, 1, 1]^\top$, where the i^{th} element of $\mathbf{N}_1$ is the multiplicity of $\mathbf{x}_i^{\text{obs}}$ in NPBS1. Similarly, NPBS2 is described by $\mathbf{N}_2 = [1, 2, 2, 0, 0, 1]^\top$ and NPBS3 by $\mathbf{N}_3 = [3, 0, 1, 1, 1, 0]^\top$. Identifying each D^{rep} with its corresponding multiplicity representation, we can then regard each nonparametric bootstrap data set as a sample from a multinomial distribution with n trials and n outcomes, $\mathbf{x}_1^{\text{obs}}, \ldots, \mathbf{x}_n^{\text{obs}}$, in which the probability of each outcome is $1/n$. That is, when performing a nonparametric bootstrap, we may consider that we are drawing samples from,

$$\text{Multinom}(n, \theta_1, \ldots, \theta_n), \tag{1.7}$$

where n is the number of trials and $\theta_i = 1/n$ is the probability associated with outcome $\mathbf{x}_i^{\text{obs}}$, for $i = 1, \ldots, n$.

1.4.1.2 Parametric bootstrap

As the name suggests, the parametric bootstrap proceeds by fitting a parametric probability model to the observations in D^{obs}, and then forming new data sets by drawing n samples from the fitted model. Usually, the model's parameters are chosen to maximise the likelihood of the observed data set. For example, if $\mathbf{x}_i^{\text{obs}} = x_i^{\text{obs}} \in \mathbb{R}$ and a univariate normal with fixed variance σ^2 is chosen as the parametric bootstrap model, then we could estimate the only parameter of our model, μ, as the sample mean of the observations. Clearly, an important consideration for our parametric model is that we should be able to sample from it. At the same time, in order for our bootstrap data samples to be realistic (in the sense that they could plausibly have been generated by the true, unknown data generating process), we must also be careful to choose a model that is not obviously in conflict with the observed data or any other knowledge/beliefs that we may have.

We note that the nonparametric bootstrap may be viewed as a special case of the parametric bootstrap, in which the chosen probability model is a multinomial (Equation 1.7) whose parameters have been selected by maximum likelihood. As we discuss in Section

1.4.3, regarding the nonparametric bootstrap in this way allows it to be considered from a Bayesian perspective.

1.4.2 Subsampling

Before moving on to Bayesian versions of the bootstrap, we consider a simple alternative for simulating data sets: namely, random subsampling. If the size of D^{obs} is n, then this approach proceeds by first choosing a value $r \in (0, n)$ and then generating data sets by repeated random sampling of r observations from D^{obs} *without* replacement. So, for example, if we return to the $n = 6$ example of Section 1.4.1.1 and take $r = 3$, then any subset of D^{obs} of size 3 might be selected. As with the nonparametric bootstrap, there is only a finite number, n_r, of distinct data sets that can be generated using subsampling approaches; namely,

$$n_r = \binom{n}{r}. \tag{1.8}$$

If we denote the set of all distinct subsets of D^{obs} of size r by $\mathcal{R}_r^{\text{obs}}$, then subsampling implicitly defines a probability model,

$$p(D^{\text{rep}}) = \begin{cases} \frac{1}{n_r}, & \text{if } D^{\text{rep}} \in \mathcal{R}_r^{\text{obs}}. \\ 0, & \text{otherwise.} \end{cases} \tag{1.9}$$

That is, random subsampling approaches sample uniformly from $\mathcal{R}_r^{\text{obs}}$. For consistency with earlier notation, we refer to the data sets generated by random subsampling as replicate data sets (even though, of course, they are of size $r < n$).

1.4.2.1 Specific applications

In the particular case where $r = n - 1$ and we are interested in estimating the bias or standard error of the summary statistic of interest, we would refer to the approach as a *jackknife* procedure (Quenouille, 1949; Tukey, 1958). Using subsampling to assess predictive performance (by training/fitting a model using the sampled set, and testing the prediction on the left-out set) is broadly known as *cross-validation*. If the training sets are generated through the subsampling procedure described above, it is common to speak of *random subsample* cross-validation, to make the distinction with k*-fold* cross-validation (where D^{obs} is first split into k subsets — or folds — and then k training data sets are defined by systematically leaving out 1 fold at a time).

1.4.2.2 Comparison to the bootstrap

One of the advantages of random subsampling compared to bootstrap approaches is that each of the replicate data sets generated by the former represents a random sample of size

r from the true, unknown DGP, F. Of course, these are not *independent* samples from F, and there will be rather more similarity between our replicate data sets than there would be between "real" data sets obtained by repeatedly drawing independent random samples of size r from F. The principal disadvantages of subsampling are that the replicate data sets are of size $r < n$, as a result of which there are fewer distinct random subsamples than there are nonparametric bootstrap samples. More formal comparisons of the asymptotic properties of subsampling and bootstrap procedures are provided by Politis and Romano (1994); Politis *et al.* (1999, 2001).

1.4.3 The Bayesian bootstrap

Recognising that there is a parametric probability model behind the bootstrap (even in the nonparametric case) allows it to be considered from a Bayesian standpoint. Although the bootstrap has its origins very firmly in the frequentist tradition, Bayesian interpretations are also possible. This was first recognised by Rubin (1981b), who introduced the *Bayesian bootstrap* (which we shall refer to as the Bayesian *nonparametric* bootstrap, for consistency with conventional bootstrap terminology).

1.4.3.1 Bayesian nonparametric bootstrap

Rather than setting the parameters in Equation (1.7) to be their maximum likelihood estimates, we may instead adopt a Bayesian approach and seek the posterior distribution, $p(\boldsymbol{\theta}|D^{\text{obs}})$, over the vector of unknown parameters, $\boldsymbol{\theta}$. We start by specifying a prior, $p(\boldsymbol{\theta})$, and then update this in light of the observed data, D^{obs}, according to Bayes rule,

$$p(\boldsymbol{\theta}|D) = \frac{p(\boldsymbol{\theta})p(D^{\text{obs}}|\boldsymbol{\theta})}{p(D^{\text{obs}})}, \tag{1.10}$$

where $p(D^{\text{obs}}|\boldsymbol{\theta})$ is the *likelihood function* (here specified by the multinomial probability model), and $p(D^{\text{obs}}) = \int_\theta p(\boldsymbol{\theta})p(D^{\text{obs}}|\boldsymbol{\theta})d\boldsymbol{\theta}$ is a normalising constant known as the *marginal likelihood*. Calculating the posterior is usually analytically intractable; however, in the case of the multinomial we may choose a conjugate Dirichlet prior, and hence determine the (Dirichlet) posterior in closed form. More concretely, if our prior is,

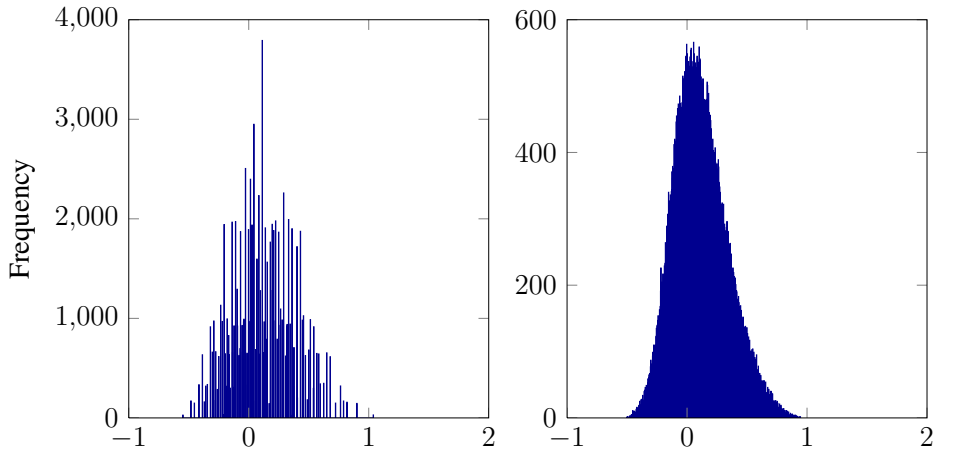
$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{n} \theta_i^{\alpha_i - 1} \tag{1.11}$$

where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^\top$ is the vector of parameters of the Dirichlet distribution and $\mathrm{B}(\boldsymbol{\alpha})$ is the multinomial Beta function, then the posterior is given by,

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}, D^{\text{obs}}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha} + \mathbf{N})} \prod_{i=1}^{n} \theta_i^{\alpha_i + N_i - 1} \tag{1.12}$$

where N_i is the multiplicity of $\mathbf{x}_i$ in the original data set D^{obs}, and $\mathbf{N} = [N_1, \ldots, N_n]^\top$ is the vector of multiplicities. Rubin (1981b) focused in particular on an improper Haldane-Dirichlet prior[1], setting $\alpha_i = 0$ for all i. The posterior corresponding to such a prior is uniform over all vectors $\boldsymbol{\theta}$ for which $\sum_{i=1}^{n} \theta_i = 1$ (and is zero otherwise).

Rubin (1981b) proposed to sample from Equation (1.12) (according to the method presented in Wilks, 1963), and considered summary statistics for which these sampled parameters could be used directly, without recourse to sampling replicate data sets. For example, if we were interested in the distribution of the mean, then we would obtain a single Bayesian bootstrap replicate by sampling $\boldsymbol{\theta}^{\text{rep}}$ from Equation (1.12), and then calculating $s^{\text{rep}} = \sum_{i=1}^{n} \boldsymbol{\theta}_i^{\text{rep}} \mathbf{x}_i^{\text{obs}}$. We would then repeat this a large number of times in order to obtain the Bayesian bootstrap distribution of s^{rep} (see Figure 1.2 (b)). Where it can be applied, this approach provides results that are rather similar to the (non-Bayesian) nonparametric bootstrap, but tends to produce smoother distributions, as illustrated in Figure 1.2. Note that this is unsurprising, since (for example) Bayesian nonparametric bootstrap replicates of the mean, $\sum_{i=1}^{n} \theta_i^* \mathbf{x}_i$, may take a continuous range of values, while conventional nonparametric bootstrap replicates are limited to values from a discrete set (albeit a potentially large one).



(a) Nonparametric bootstrap (b) Bayesian nonparametric bootstrap

Figure 1.2: We seek the bootstrap distribution of the sample mean of 5 numbers drawn from a standard normal. These are: 1.04, 0.35, -0.21, -0.07, -0.55. (a) The distribution of the sample mean determined from 10,000 nonparametric bootstrap samples. (b) The distribution of $s^{\text{rep}} = \sum_{i=1}^{n} \boldsymbol{\theta}_i^{\text{rep}} \mathbf{x}_i^{\text{obs}}$ determined from 10,000 Bayesian nonparametric bootstrap samples (employing a Haldane-Dirichlet prior as in Rubin 1981b). The similarity between the two distributions is largely due to the use of a non-informative prior in the Bayesian case, as discussed in Section 8.4 of Hastie *et al.* (2009).

[1] A prior density, $p(\boldsymbol{\theta})$, is called *improper* if the integral $\int p(\boldsymbol{\theta}) d\boldsymbol{\theta}$ diverges (see, for example, Gelman, 2004). Such priors can still be useful, provided the posterior $p(\boldsymbol{\theta}|D^{\text{obs}})$ is not also improper. We note that the Haldane-Dirichlet prior cannot strictly be written as in Equation (1.11), which is only valid when $\alpha_i > 0$ for all i. In this case, Equation (1.11) should more correctly be written as $p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \propto \prod_{i=1}^{n} \theta_i^{\alpha_i - 1}$. Equation (1.12) remains valid, provided $N_i \geq 1$ for all i.

One of the key differences between the conventional (frequentist) and Bayesian versions of the nonparametric bootstrap is therefore that the former considers summaries of the form $s = T(D)$, while the latter considers summaries $s = T(D^{\text{obs}}, \boldsymbol{\theta})$. As previously discussed, for the frequentist nonparametric bootstrap, the variability in s follows as a direct consequence of the variability in D. Our objective is hence to find the sampling distribution of s. The form $s = T(D)$ reflects these assumptions: D is a random variable (whose distribution is unknown, and which we therefore approximate) and s is simply a deterministic function of D. In contrast, for the Bayesian nonparametric bootstrap, the variability in s follows from the uncertainty in the unknown parameter $\boldsymbol{\theta}$. Hence, in this case, the form $s = T(D^{\text{obs}}, \boldsymbol{\theta})$ is appropriate: the observed data set, D^{obs}, is fixed and s is a deterministic function of this fixed set and the random variable $\boldsymbol{\theta}$. It follows that the Bayesian nonparametric bootstrap distribution may be interpreted as the posterior distribution of s, given the multinomial model and the prior $p(\boldsymbol{\theta})$ (Rubin, 1981b). We shall refer to this as the *induced* posterior distribution of s.

We note that we could, in principle, generate replicate data sets using the Bayesian bootstrap, by sampling from the joint distribution,

$$p(D^{\text{rep}}, \boldsymbol{\theta} | D^{\text{obs}}) = p(D^{\text{rep}} | \boldsymbol{\theta}, D^{\text{obs}}) p(\boldsymbol{\theta} | D^{\text{obs}}). \tag{1.13}$$

As we discuss in the next section, this is exactly the approach taken when performing *posterior predictive checks*.

1.4.4 Posterior predictive checking

A Bayesian procedure that is in many ways similar to the bootstrap is *posterior predictive checking* (Rubin, 1984; Gelman *et al.*, 1996; Gelman, 2004), also known as *phenomenological Bayesian monitoring* (Rubin, 1981a). We provide details of this here (largely following the notation of Gelman, 2004).

We suppose that we have an observed data set, D^{obs}, and a parametric probability model, $\hat{F}$, for the DGP, whose parameters are denoted $\boldsymbol{\theta}$. As previously, we suppose that — given a realisation of the parameters — we are able to simulate from the model in order to generate replicate data sets. For consistency with previous sections, we here assume that the simulated data sets always comprise the same number of samples as the original data set (we refer to Gelman *et al.*, 1996, for a more general treatment, where arbitrary properties of D^{obs} — not just the sample size — may be maintained). Posterior predictive checking is an approach for assessing the quality of the model, $\hat{F}$, through simulation. In the first stage of this procedure, we seek to sample replicate data sets from the *posterior*

predictive distribution,

$$p(D^{\text{rep}}|D^{\text{obs}},\hat{F}) = \int p(D^{\text{rep}},\boldsymbol{\theta}|D^{\text{obs}},\hat{F})d\boldsymbol{\theta} \tag{1.14}$$

$$= \int p(D^{\text{rep}}|\boldsymbol{\theta},D^{\text{obs}},\hat{F})p(\boldsymbol{\theta}|D^{\text{obs}},\hat{F})d\boldsymbol{\theta} \tag{1.15}$$

$$= \int p(D^{\text{rep}}|\boldsymbol{\theta},\hat{F})p(\boldsymbol{\theta}|D^{\text{obs}},\hat{F})d\boldsymbol{\theta}, \tag{1.16}$$

where $p(\boldsymbol{\theta}|D^{\text{obs}},\hat{F})$ is the posterior distribution of $\boldsymbol{\theta}$, and the final line follows from the property that D^{rep} and D^{obs} are conditionally independent of one another, given $\boldsymbol{\theta}$ and $\hat{F}$. In order to generate replicate data sets, we sample from the joint distribution, $p(D^{\text{rep}},\boldsymbol{\theta}^{\text{rep}}|D^{\text{obs}},\hat{F})$, by first sampling $\boldsymbol{\theta}^{\text{rep}}$ from $p(\boldsymbol{\theta}|D^{\text{obs}},\hat{F})$, and then simulating from $\hat{F}$. We repeat this a large number, B, of times and hence obtain a set of pairs, $\{(\boldsymbol{\theta}_i^{\text{rep}},D_i^{\text{rep}})\}_{i=1}^B$. Equation (1.16) tells us that the replicate data sets, D_i^{rep}, generated in this way constitute samples from the posterior predictive distribution.

To perform the posterior predictive check, we must also define one or more *test statistics*, $T(D)$, which are scalar summaries of the data (or, more generally, we define *discrepancy measures*, $T(D,\boldsymbol{\theta})$, which are summaries of both the data *and* the parameters; see Gelman *et al.*, 1996). We may then calculate $T(D_i^{\text{rep}})$ for each of the replicate data sets (or, if we are using a discrepancy measure, we calculate $T(D_i^{\text{rep}},\boldsymbol{\theta}_i^{\text{rep}})$). This provides us with a frequency distribution for the test statistic. As in the Bayesian nonparametric bootstrap case, this represents the induced posterior distribution of T. Posterior predictive checking refers to any procedure by which $T(D^{\text{obs}})$ (the test statistic evaluated on the original data set) is compared to this distribution. Informally, if $\hat{F}$ is a "good" model, then $T(D^{\text{obs}})$ will be a "typical" value for T. This may be quantified by considering whether or not $T(D^{\text{obs}})$ appears in the tails of the induced posterior predictive distribution, and gives rise to the concept of a *Bayesian p-value* (Gelman *et al.*, 1996; Gelman, 2004).

As a model-checking procedure, posterior predictive checks have been criticised (Bayarri and Berger, 1999). The main difficulty arises from using the data twice: once to obtain the posterior, $p(\boldsymbol{\theta}|D^{\text{obs}},\hat{F})$, and then again when comparing $T(D^{\text{obs}})$ to the induced posterior distribution of the test statistic. The result of this is that posterior predictive checks tend to be conservative (i.e. they are not as critical of the model being checked as perhaps they should be). One well-established alternative is the *prior predictive check* (Box 1980; see also Ratmann *et al.* 2009 for a recent novel application in the context of approximate Bayesian computation). In this approach, we replace the posterior, $p(\boldsymbol{\theta}|D^{\text{obs}},\hat{F})$, in Equation (1.16) by the prior, $p(\boldsymbol{\theta}|\hat{F})$. This clearly eliminates the problem of using the data twice, but introduces a number of practical difficulties (Bayarri and Berger, 1999). Chief amongst these is the strong dependence upon the choice of prior: a good model may nevertheless appear to be at odds with the data if a poor prior is used. A number of alternatives to the prior and posterior checking procedures have been proposed, which seek to avoid using the data twice and also to diminish the effects of a poorly chosen prior. For example, cross-validation approaches may be used, in which a subset of the data is used in order to find the posterior, and then the remainder is used when performing the

'check' (Carlin, 1999; Marshall and Spiegelhalter, 2003). In the context of hierarchical models, *mixed predictive checks* offer another alternative (Gelman *et al.*, 1996; Marshall and Spiegelhalter, 2007; Lewin *et al.*, 2007). In this case, we consider priors, $p(\boldsymbol{\theta}|\boldsymbol{\beta}, \hat{F})$, that are themselves parameterised by *hyperparameters*, $\boldsymbol{\beta}$. Samples, $\boldsymbol{\beta}^{\text{rep}}$, are drawn from the "hyper-posterior", $p(\boldsymbol{\beta}|D^{\text{obs}}, \hat{F})$, which are then plugged into the prior. In turn, samples, $\boldsymbol{\theta}^{\text{rep}}$, are then drawn from the prior, $p(\boldsymbol{\theta}|\boldsymbol{\beta}^{\text{rep}}, \hat{F})$, after which replicate data sets may be obtained by simulating from $\hat{F}$. Although mixed predictive checks do use the data more than once, they have been found to be less conservative than their posterior predictive counterparts (Marshall and Spiegelhalter, 2003).

Finally, we note that the use of predictive checks and (in particular) p-values is not universally accepted within the Bayesian paradigm (see, for example Lindley, 1999). Although an interesting source of debate, we do not enter into this discussion here, and turn instead to our principal concern: the connection of the posterior predictive checking procedure to the bootstrap.

1.4.4.1 Comparison to the bootstrap

The machinery of posterior predictive checking may be viewed as a generalisation of both the frequentist and Bayesian bootstrap procedures. For example, if the posterior, $p(\boldsymbol{\theta}|D^{\text{obs}}, \hat{F})$, in Equation (1.16) were a spike (delta function) located at $\boldsymbol{\theta}_{ML}$ (the maximum likelihood parameter vector), then the replicate data sets generated by simulating from the posterior predictive distribution would be identical to (frequentist) parametric bootstrap samples (we note that this observation has previously been made by Bollback, 2005). On the other hand, if we were to choose a discrepancy measure $T(D, \boldsymbol{\theta})$ that did not vary with D, but was instead a function of the unknown $\boldsymbol{\theta}$ and the observed data set D^{obs}, then (in the particular case where $\hat{F}$ were a multinomial distribution and the prior on $\boldsymbol{\theta}$ were a conjugate Dirichlet distribution) we would recover the Bayesian nonparametric bootstrap.

The aims of posterior predictive checking and the bootstrap are quite different, however. In the former, we seek to assess the quality of the approximation $\hat{F}$ by comparing summaries of the replicate data sets (the tests statistics) with the same summaries of the original data set. In the latter, although we might perform checks to determine that $\hat{F}$ is an adequate model of the DGP, our principal aim is to assess the variability of summaries derived from the replicate data sets. However, we note that there is no reason why the induced distribution of T that is constructed during the posterior predictive checking procedure should not be used in order to make statements about the stability of the test statistic (once we have established that $\hat{F}$ is not a poor model).

1.5 Alternative Bayesian method for stability assessment

So far, we have only discussed methods that assess stability by first approximating the DGP, which is the approach that we shall take throughout this thesis. An alternative approach that we might consider is to perform Bayesian inference directly upon the unknown quantity whose stability we wish to assess (which, in this context — and for reasons that will become clear below — we will denote by ϕ rather than s). We would then seek the posterior distribution $p(\phi|D^{\text{obs}}, \mathcal{H})$, where we here use $\mathcal{H}$ as a catchall for any modelling assumptions that are made. Rather than assessing stability *per se*, we would instead be quantifying the uncertainty that remains in the value of ϕ after having observed D^{obs}.

1.5.1 Comparison to methods of Section 1.4

Recall that, throughout Section 1.4, we were interested in methods that could generate replicate data sets, D^{rep}, from an approximation, $\hat{F}$, to the DGP. When $\hat{F}$ was parametric, we either performed maximum likelihood estimates of the parameters (in the case of the "conventional" bootstrap), or sought the posterior distribution, $p(\boldsymbol{\theta}|D^{\text{obs}}, \hat{F})$ (in the cases of the Bayesian bootstrap and posterior predictive checking procedure). We calculated summary statistics, s, that were deterministic functions of the replicate data set, D^{rep}; and/or the parameters, $\boldsymbol{\theta}$; and possibly also the original data set, D^{obs}. To cover all of these possibilities, we shall here write $s = T(D^{\text{rep}}, \boldsymbol{\theta}, D^{\text{obs}})$, where T is a deterministic function. As previously described, the (induced) posterior distribution of the summary statistic is then,

$$\begin{aligned}
&p(T(D^{\text{rep}}, \boldsymbol{\theta}, D^{\text{obs}})|D^{\text{obs}}, \hat{F}) \\
&\quad = \int\int p(T(D^{\text{rep}}, \boldsymbol{\theta}, D^{\text{obs}}), D^{\text{rep}}, \boldsymbol{\theta}|D^{\text{obs}}, \hat{F}) dD^{\text{rep}} d\boldsymbol{\theta} \\
&\quad = \int\int p(T(D^{\text{rep}}, \boldsymbol{\theta}, D^{\text{obs}})|D^{\text{rep}}, \boldsymbol{\theta}, D^{\text{obs}}) p(D^{\text{rep}}|\boldsymbol{\theta}, \hat{F}) p(\boldsymbol{\theta}|D^{\text{obs}}, \hat{F}) dD^{\text{rep}} d\boldsymbol{\theta} \\
&\quad = \int\int T(D^{\text{rep}}, \boldsymbol{\theta}, D^{\text{obs}}) p(D^{\text{rep}}|\boldsymbol{\theta}, \hat{F}) p(\boldsymbol{\theta}|D^{\text{obs}}, \hat{F}) dD^{\text{rep}} d\boldsymbol{\theta}, \qquad (1.17)
\end{aligned}$$

where the final line is a consequence of the fact that T is deterministic.

It follows from Equation (1.17) that a sample from the induced posterior may be obtained by,

1. drawing a sample, $\boldsymbol{\theta}_i^{\text{rep}}$, from $p(\boldsymbol{\theta}|D^{\text{obs}}, \hat{F})$;
2. drawing a sample, D_i^{rep}, from $p(D^{\text{rep}}|\boldsymbol{\theta}_i^{\text{rep}}, \hat{F})$; and
3. calculating $s_i^{\text{rep}} = T(D_i^{\text{rep}}, \boldsymbol{\theta}_i^{\text{rep}}, D^{\text{obs}})$.

This process precisely captures the methodology of the posterior predictive checking procedure, the Bayesian nonparametric bootstrap, and also — if $p(\boldsymbol{\theta}|D^{\text{obs}}, \hat{F})$ is approximated by a spike at $\boldsymbol{\theta}_{ML}$ — the conventional bootstrap (see also Section 1.4.4.1).

All of the methods of Section 1.4 may therefore be viewed as generating samples from $p(s|D^{\text{obs}}, \hat{F})$ (or approximations to it), and hence are similar to the Bayesian approach that we outlined at the start of this section (which seeks $p(\phi|D^{\text{obs}}, \hat{F})$). The principal distinction stems from the nature of s, which is assumed to be deterministically specified given D and/or $\boldsymbol{\theta}$. It is this assumption that allows us to derive the posterior distribution of s by performing inference on D and $\boldsymbol{\theta}$. For more general quantities, ϕ, such an approach would not be possible. Given the aims of this thesis, this is not too much of a limitation. In particular, we may consider any quantity, s, that is calculated deterministically from a data set using a computer program. This allows us to assess the stability of the outputs of pre-existing (and often complex) computer programs simply by altering their inputs.

1.6 Thesis overview and outline

In Chapter 2 we consider the use of stability selection methods for identifying robust biomarkers. We propose a novel algorithm that combines assessments of stability and predictive performance, which is then applied to a simulation example. In Chapter 3 we consider the particular biological example of discovering protein peak biomarkers of the inflammatory condition, HTLV-1 associated myelopathy/tropical spastic paraparesis (HAM/TSP). We select a number of putative biomarkers, two of which have been experimentally identified. We then move on to consider time courses of data. In Chapter 4 we propose a method for bootstrapping such data sets, and then apply this approach in Chapters 5 and 6. We first use the procedure to quantify the stability of networks inferred from gene expression time course data (Chapter 5), and then consider the stability of parameter estimates for ordinary differential equation models (Chapter 6). Finally, in Chapter 7 we summarise and discuss the main conclusions of this thesis.

Chapter 2

Stability selection for biomarker discovery

Abstract *Recent studies have highlighted the importance of assessing the stability with which putative biomarkers are selected. We here present a generic method for selecting covariates (i.e. putative biomarkers) on the strength of a simple probabilistic score that combines assessments of stability and diagnostic performance. By applying this score in tandem with a selection strategy based upon the elastic net, we assess the effects of correlations upon stability.*

Outline *In Section 2.1, we provide an introduction to the biomarker discovery problem, after which we review in Section 2.2 existing methods for assessing selection stability. We then describe the feature selection strategies that we employ (Section 2.3), before discussing the "stability selection" procedure that was recently proposed by Meinshausen and Bühlmann (2010). We highlight certain difficulties with this procedure that can arise as a result of correlations amongst covariates. In Section 2.5 we present a novel strategy for selecting covariates on the basis of both stability and predictive performance. We discuss the implementation of this approach in Section 2.6. In Section 2.7, we consider a simulation example that allows us to demonstrate the utility of our algorithm relative to a more conventional approach that selects solely on the basis of predictive performance. We discuss our results in Section 2.9*

The principal motivation for this work is a real biological example, in which we seek SELDI mass spectrometry protein peak biomarkers of HTLV-1 associated myelopathy. We describe this example fully in the next chapter, where we also provide a detailed analysis of the data using both our novel selection method, as well as more traditional approaches.

2.1 Background

The search for molecular profiles that allow us to discriminate between two (or more) classes of individuals or entities is an ongoing challenge in genomic, metabolomic and proteomic studies. Such profiles may simply be sought as diagnostic or prognostic aids (e.g. van de Vijver *et al.*, 2002; van 't Veer *et al.*, 2002), or may represent a first step toward understanding a given biological process (e.g. Vine *et al.*, 2004). The biomolecules that appear in these profiles are termed *biomarkers*, and the process of identifying them is known as *biomarker discovery*.

2.1.1 Differential expression analyses

One important example of a biomarker discovery procedure is *differential expression* (see also Section 1.2.2). Here, we seek to identify genes or proteins that are up- or down-regulated in a collection of "case" samples relative to a collection of "controls". The cases and controls may be individuals who do or do not suffer from a disease, or may represent more general binary classifications (such as mutant versus wildtype; subjected to stress versus grown in ideal conditions; or any one of a wide array of different dichotomous relationships). Typically, DNA microarrays or other high-throughput technologies are used, which allow expression levels to be measured upon hundreds or thousands of genes or proteins simultaneously. The development of statistical procedures for detecting which genes or proteins are differentially expressed between the classes of interest remains an area of intense research, often driven by the emergence of new high-throughput technologies (Bullard *et al.*, 2010; Robinson *et al.*, 2010; Stegle *et al.*, 2010).

The methods for differential expression analyses tend to be univariate gene-by-gene (or protein-by-protein) approaches that employ t-tests or alternatives/variants thereof (including nonparametric and empirical Bayes procedures; see, for example, Efron *et al.*, 2001). Given the large number of genes or proteins that usually appear in these studies, correcting for multiple comparisons is essential (see Section 1.2.2). At the end of a differential expression analysis, we are typically presented with a list of genes/proteins (often ranked by p-value) that we have determined to be significantly differentially expressed between the two classes/conditions. The entries of this list represent the putative biomarkers. The main limitations of such univariate approaches is that they do not take into account possible interactions or dependencies *amongst* the covariates, and also that they do not by themselves allow predictions to be made. For these reasons, it has become increasingly common to consider multivariate predictive models.

2.1.2 Multivariate predictive models for biomarker discovery

Multivariate predictive models that select covariates (in our case, putative biomarkers) may be broadly classified as either *filter*, *wrapper* or *embedded* procedures. Filter methods separate out the feature selection and prediction tasks, making use of an initial *filtering* step in order to identify relevant covariates (possibly using univariate procedures), and then training a multivariate classifier using the reduced set of features (Guyon and Elisseeff, 2003). In contrast, *wrapper* approaches combine the two tasks by searching through different feature sets and assessing each one by fitting the corresponding model to the observed data (Kohavi and John, 1997). Finally, *embedded* procedures refer to multivariate models which automatically select features as part of the process of fitting to an observed data set (Saeys *et al.*, 2007). In general, filter methods provide the worst predictive performance (as we might expect, since the selected features are chosen independently of the predictive model), while wrapper methods are usually the most computationally costly, as they require a search through a large number of different combinations of the covariates, with the predictive model needing to be fitted for each one (Saeys *et al.*, 2007). Embedded methods often offer similar levels of predictive performance to wrapper approaches, while maintaining computational efficiency (Guyon and Elisseeff, 2003).

For the sake of brevity and focus, we do not provide a review of the many different feature selection methods that are employed in bioinformatics and systems biology, and refer to Guyon and Elisseeff (2003) and, particularly, Saeys *et al.* (2007) for comprehensive treatments. In Section 2.3, we describe the method that we use throughout this chapter and the next, which is an embedded strategy employing a logistic regression classifier with a lasso or elastic net likelihood penalty. However, many of the concepts that we discuss are generic, rather than being specific to any particular choice of selection strategy.

2.1.3 Selection stability

A number of recent papers have highlighted the importance of considering the stability of covariates identified by feature selection algorithms, especially in the context of identifying gene signatures and biomarkers of disease (for example, Abeel *et al.*, 2010; Meinshausen and Bühlmann, 2010; Zucknick *et al.*, 2008). The principal aim is to establish how specific the covariates selected (i.e. the identified biomarkers) are to the particular data set that was observed, in order to quantify how well we might expect our selections to generalise to new data sets. Although not a new concept (see, for example, Turney, 1995, for an early discussion), it has received a renewed interest in biological contexts due to concerns over the irreproducibility of results (Ein-Dor *et al.*, 2006, 2005). Assessments of stability usually proceed by subsampling the original data set. A feature selection algorithm is applied to each subsample, and then stability is quantified using any one of several methods for assessing the concordance amongst the resulting sets of selections (as described in Section 2.2). There is an increasing body of literature on this subject, and we refer to He and Yu (2010) for an alternative review to the one presented here.

One of the main difficulties with stability is that it is not by itself a useful objective. As pointed out in Abeel *et al.* (2010), a selection strategy that always chooses a fixed set of covariates regardless of the observed data will provide perfect stability, but the predictive performance is likely to be poor. Since we ultimately seek biomarkers that are not only robust but which also allow us to discriminate between (for example) different disease states, it is desirable to try to optimise both stability and predictive performance at the same time. The procedure of Abeel *et al.* (2010) addresses this by taking a powerful predictive model with an embedded covariate selection strategy (a *Support Vector Machine* using the *Recursive Feature Elimination* method— see Guyon *et al.*, 2002), and then using a bootstrap aggregation approach in order to improve selection stability. In Section 2.5, we present a simple alternative methodology. We follow Meinshausen and Bühlmann (2010) in estimating selection probabilities for different sets of covariates, but diverge from their approach by combining these estimates with assessments of predictive performance. Given that our approach uses subsampling for both model structure estimation and performance assessment, it is somewhat related to double cross validation (see Stone, 1974, and also Smit *et al.*, 2007 for a proteomics application similar to the one considered in the next chapter); however, we do not employ a nested subsampling step.

A further difficulty with stability arises as a result of correlation. As discussed in Yu *et al.* (2008), Kirk *et al.* (2010) and Section 2.4.2, correlations amongst covariates can have a serious impact upon stability. Since multivariate covariate selection strategies commonly seek a minimal set of covariates that yield the best predictive performance, a single representative from a group of correlated covariates is often selected in favour of the whole set. This can have a negative impact upon stability, as the selected representative is liable to vary from subsampled data set to subsampled data set. Unfortunately, due to complex interdependencies between biological entities, correlations are ubiquitous in genomic and proteomic studies. A further contribution of this chapter is therefore to quantify the effects of correlation upon stability. The covariate selection strategy that we ultimately employ provides a parameter, α, that can be varied in order to control whether we tend to select single representatives or whole sets of correlated covariates (see Zou and Hastie, 2005; Friedman *et al.*, 2007, 2010). This allows us to investigate systematically how our treatment of correlation affects stability.

2.2 Assessing stability

There are various ways of quantifying the stability with which variables are selected. After first defining some notation, in this section we consider the Jaccard and Kuncheva similarity indices, as well as estimated selection probabilities.

2.2.1 Notation

As in Chapter 1, let D^{obs} be the data set of n observations, where now $D^{\text{obs}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$. Here, each $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]^\top \in \mathbb{R}^p$ is a vector of measurements taken upon p covariates, $v_1, \ldots, v_p$, and each $y_i \in \{0, 1\}$ is the corresponding observation of the class label, y. We suppose that we have a selection strategy, $\mathcal{H}$, that seeks to find a set comprising the "best" covariates (the definition of "best" is implicitly encoded in the particular choice of strategy). We further suppose that this strategy has an associated parameter, λ, which provides control over the number of selections made. For data set, D, and parameter, λ, we denote the set of covariates selected by $\mathcal{H}$ as $s(D; \lambda, \mathcal{H})$. To assess *selection stability*, we would ideally investigate the consistency amongst sets $s(D_i; \lambda, \mathcal{H})$, where each D_i is a draw from the underlying data generating process, F. Since it is usually infeasible to obtain large numbers of independent data sets, a practical way in which to assess stability is to consider subsampling or bootstrapping D^{obs}, as described in Chapter 1. Applying the plug-in principle, we consider the consistency amongst sets $s(D_i^{\text{rep}}; \lambda, \mathcal{H})$, where each D_i is a draw from an approximating DGP, $\hat{F}$.

2.2.2 Jaccard similarity coefficient

The Jaccard coefficient (Jaccard, 1901, 1912) was introduced as a way of measuring the similarity between two sets S_1 and S_2,

$$J(S_1, S_2) := \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \tag{2.1}$$

Note that J attains a minimal value of 0 if and only if $S_1 \cap S_2 = \emptyset$, and a maximal value of 1 if and only if $S_1 = S_2$.

To provide a summary of the similarity between several sets $S_1, \ldots, S_\ell$, we may consider the mean pairwise Jaccard coefficient, $\overline{J}$, where,

$$\overline{J}(S_1, \ldots, S_\ell) := \frac{2}{\ell(\ell-1)} \sum_{i=1}^{\ell-1} \left(\sum_{j=i+1}^{\ell} J(S_i, S_j) \right). \tag{2.2}$$

Since $\overline{J}$ is defined as the mean of $\ell(\ell-1)/2$ Jaccard coefficients, it follows that $\overline{J}$ also has a maximum of 1 and a minimum of 0. Furthermore, the maximum occurs only when all S_i are identical, and the minimum occurs only when all S_i are pairwise disjoint (i.e. when $S_i \cap S_j = \emptyset$ for all i and j).

Calculating the mean pairwise Jaccard coefficient, $\overline{J}$, for the selected sets $s(D_1^{\text{rep}}; \lambda, \mathcal{H})$, $\ldots, s(D_B^{\text{rep}}; \lambda, \mathcal{H}))$ provides us with an assessment of selection stability (Zucknick *et al.*, 2008). We would regard $\overline{J} = 1$ as corresponding to maximal stability (complete agreement amongst the selections) and $\overline{J} = 0$ as corresponding to maximal instability (complete disagreement amongst the selections) .

We note that some recent authors refer to the mean pairwise Jaccard coefficient, $\overline{J}$, as the *generalised Kalousis measure* (Somol and Novovičová, 2010), following the use by Kalousis *et al.* (2007) of an "adaptation of the Tanimoto distance" (see also Tanimoto, 1960). However, this adaptation is in fact identical to the Jaccard coefficient, so we persist with our slightly more wordy terminology for the sake of clarity.

2.2.3 Kuncheva similarity coefficient

To motivate the exposition that follows, we first consider the behaviour of the Jaccard coefficient under a random selection strategy. Suppose that S_1 and S_2 are both random subsets of the complete set of covariates, $\{v_1, \ldots, v_p\}$. Furthermore, suppose S_1 and S_2 are both of size m, with $0 < m < p$. On average, how many elements will S_1 and S_2 have in common? An alternative way of phrasing this is as follows: if the elements of S_1 are all considered as "successes", and the remaining elements of $\{v_1, \ldots, v_p\}$ are considered as "failures", how many successes will appear in S_2 (on average)? Having phrased the problem in this way, it is clear that the solution is given by the expectation of a hypergeometric distribution with m draws, m successes, and population size p, and hence is m^2/p (Feller, 1968).

From the above, it follows that, if we were to adopt a selection strategy that simply draws m covariates at random (where $0 < m < p$), the expectation of the Jaccard coefficient is given by,

$$\begin{aligned} E[J(S_1, S_2)] &= E\left[\frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}\right] \\ &= E\left[\frac{|S_1 \cap S_2|}{|S_1| + |S_2| - |S_1 \cap S_2|}\right] \\ &= \frac{m^2/p}{2m - m^2/p} \\ &= \frac{1}{2(p/m) - 1}. \end{aligned} \tag{2.3}$$

Hence, under a random selection strategy, the Jaccard coefficient will tend to increase as the number of selections, m, increases.

Kuncheva (2007) defines an alternative similarity coefficient that corrects for the increase in selection stability that occurs as a result of random chance. If $|S_1| = |S_2| = m$, then

$$K(S_1, S_2) := \frac{|S_1 \cap S_2|p - m^2}{mp - m^2}, \tag{2.4}$$

where p is the total number of covariates. The expectation of this similarity coefficient under a random selection strategy is zero for all $m \in (0, p)$. Moreover, its maximum value is 1 (when $S_1 = S_2$) and its minimum is -1 (when $S_1 \cap S_2$ and $m = p/2$).

The mean pairwise Kuncheva coefficient, $\overline{K}$, can be calculated in a similar manner to the mean pairwise Jaccard coefficient, simply exchanging J for K in Equation (2.2).

2.2.4 Other similarity measures

There are many other similarity measures that may be used to assess selection stability, including (to name but a few): the relative Hamming distance (Dunne *et al.*, 2002); the Dice-Sørensen index (Dice, 1945; Sørensen, 1948); and the Ochiai index (Ochiai, 1957). An extensive review of these and others is provided in He and Yu (2010). All of these indices have commonalities, and there are often only subtle differences between them. We consider only the Jaccard and Kuncheva similarity indices. The Jaccard index is particularly popular in the literature (e.g. Zucknick *et al.*, 2008; Kalousis *et al.*, 2007; Saeys *et al.*, 2008), while the more recent Kuncheva index is considered to be a particularly suitable measure for stability analyses due to its correction for agreement due to random chance (Abeel *et al.*, 2010).

2.2.5 Selection probabilities

Meinshausen and Bühlmann (2010) introduce a rather different way of assessing stability, in the form of *estimated selection probabilities*. For any subset, V, of the set of covariates $\{v_1, \ldots, v_p\}$, they consider the probability that V is selected by strategy, $\mathcal{H}$. The way to estimate this probability in the frequentist framework would be to sample data sets, $D_1, \ldots, D_N$, from the DGP, F; to apply $\mathcal{H}$ to sampled data set; and finally to calculate the proportion of times that V is selected. As $N \to \infty$ this proportion tends to the (frequentist) probability of selecting V. A straightforward application of the plug-in principle tells us to estimate this probability as,

$$\widehat{\mathrm{P}}_{\mathrm{Sel}}(V|\lambda, \mathcal{H}) := \frac{1}{B} \sum_{i=1}^{B} \mathbb{I}\left(V \subseteq s(D_i^{\mathrm{rep}}; \lambda, \mathcal{H})\right), \tag{2.5}$$

where $D_1^{\mathrm{rep}}, \ldots, D_B^{\mathrm{rep}}$ are replicate data sets drawn from the approximate DGP, $\hat{F}$. When calculated for sets V such that $|V| = 1$ (i.e. sets that comprise just a single covariate), we refer to these estimates as *marginal* selection probabilities.

We note that this method of stability assessment is conceptually different to both the Jaccard and Kuncheva coefficients of Sections 2.2.2 and 2.2.3. Instead of providing a global summary of the variability amongst a collection of selected sets, the method of Meinshausen and Bühlmann (2010) associates a "stability score" (interpretable as an estimated probability) with every subset of the covariates. The higher the score, the more stably selected the covariate. As we discuss in Section 2.4, one way to proceed is to place a cutoff on this score in order to identify a set of the most stably selected covariates. Before that, however, it will prove useful to introduce two specific feature selection methods.

2.3 Feature selection algorithms

There is a wide variety of different feature selection strategies, many of which are reviewed in Saeys *et al.* (2007). We here focus on *penalised likelihood methods* applied to logistic regression models for binary classification. We first introduce logistic regression, and then describe the lasso and elastic net penalties.

2.3.1 Logistic regression

The standard logistic regression model for the binary classification problem is as follows,

$$p(y = 1|\mathbf{v}^\top = \mathbf{z}^\top; \beta_0, \boldsymbol{\beta}) = f(\beta_0 + \boldsymbol{\beta}^\top \mathbf{z}), \tag{2.6}$$

where $\mathbf{v} = [v_1, \ldots, v_p]^\top$ is the vector of the covariates; $\mathbf{z} = [z_1, \ldots, z_p]^\top$ is a corresponding vector of observed values; $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^\top$ is a vector of coefficients; β_0 is an intercept term; and f is the logistic function,

$$f(t) = \frac{t}{1 + \exp(-t)}. \tag{2.7}$$

Given a data set, D^{obs}, the usual way to estimate the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ is to identify the values that maximise the (log) likelihood function. Recall that $D^{\text{obs}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]^\top \in \mathbb{R}^p$, and $y_i \in \{0, 1\}$. Assuming independent observations, the log likelihood is given by,

$$\ell(\beta_0, \boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i \log \left(p(y_i|\mathbf{x}_i; \beta_0, \boldsymbol{\beta}^\top) \right) + (1 - y_i) \log \left(1 - p(y_i|\mathbf{x}_i; \beta_0, \boldsymbol{\beta}^\top) \right) \right), \tag{2.8}$$

where $p(y_i|\mathbf{x}_i; \beta_0, \boldsymbol{\beta})$ is shorthand for $p(y_i = 1|\mathbf{v}^\top = \mathbf{x}_i; \beta_0, \boldsymbol{\beta})$, which is calculated by plugging the observed data into Equation (2.6).

The β_i's that maximise the log likelihood may be found by setting the derivatives of Equation (2.8) to be zero, and then solving the resulting equations using (for example) the Newton-Raphson algorithm (Press *et al.*, 2007).

2.3.2 The lasso

The lasso (Tibshirani, 1996; Efron *et al.*, 2004) introduces a penalty term, $\sum_{i=1}^p |\beta_i|$, to the likelihood function, so that the estimated β_i's are now given by,

$$\widehat{\beta}_0, \widehat{\boldsymbol{\beta}} = \underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmax}} \left[\ell(\beta_0, \boldsymbol{\beta}) - \lambda \sum_{i=1}^{p} |\beta_i| \right], \tag{2.9}$$

where $\lambda \geq 0$ is a *regularisation parameter* which controls the strength (severity) of the penalty. When $\lambda = 0$, we recover the unpenalised form of the likelihood. For λ sufficiently large (say, $\lambda \geq \lambda_{\max}$), the maximum value of the penalised likelihood will be given by setting $\beta_i = 0$, for $i = 1, \ldots, p$. It is clear that the general effect of the penalty term will be to shrink the magnitude of the estimated coefficients. However, the real benefit of the lasso is that, rather than simply shrinking all of the coefficients to be small, some of the coefficients are set exactly to zero (Tibshirani, 1996). As a result of this, the lasso performs automatic feature selection, where the covariates with non-zero coefficients constitute the selected features.

2.3.2.1 The regularisation parameter

The value of λ determines how many covariates are selected: the larger λ, the fewer selections. Choosing the "best" value for λ is a non-trivial problem, and often we resort to cross-validation approaches in order to find the λ which provides the lowest generalisation error (Hastie *et al.*, 2009).

Rather than specifying a single value for λ, we may be interested in determining how the estimated coefficients change as λ is reduced from $\lambda_{\max}$ to 0. This information is often presented as a *regularisation path*, as illustrated in Figure 2.1. Several approaches now exist for finding the regularisation path (e.g. Friedman *et al.*, 2010; Koh *et al.*, 2007; Genkin *et al.*, 2007). Throughout this thesis, we employ the freely available `glmnet` package in `R`, which implements the method of Friedman *et al.* (2010).

2.3.3 The elastic net

An alternative to the lasso is provided by the *elastic net* (Zou and Hastie, 2005). This is again a penalised likelihood approach, and may be viewed as a generalisation of the lasso. For the elastic net, the estimated β_i's are given by,

$$\widehat{\beta}_0^{(EN)}, \widehat{\boldsymbol{\beta}}^{(EN)} = \underset{\beta_0, \boldsymbol{\beta}}{\operatorname{argmax}} \left[\ell(\beta_0, \boldsymbol{\beta}) - \lambda Q_\alpha(\boldsymbol{\beta}) \right], \tag{2.10}$$

where,

$$Q_\alpha(\boldsymbol{\beta}) = \sum_{j=1}^{p} \left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right], \tag{2.11}$$

is the elastic net penalty term[1]. Note that this penalty term introduces a second parameter, α, in addition to λ. When $\alpha = 1$, we recover the lasso (Equation 2.9), and when $\alpha = 0$ we recover the *ridge* penalty (Hoerl, 1962). On its own, the ridge penalty has the effect of

[1] Often the definition of the elastic net penalty term omits the factor of 1/2 that multiplies the first term in the sum (e.g. Hastie *et al.*, 2009). We keep this factor for consistency with Friedman *et al.* (2010).

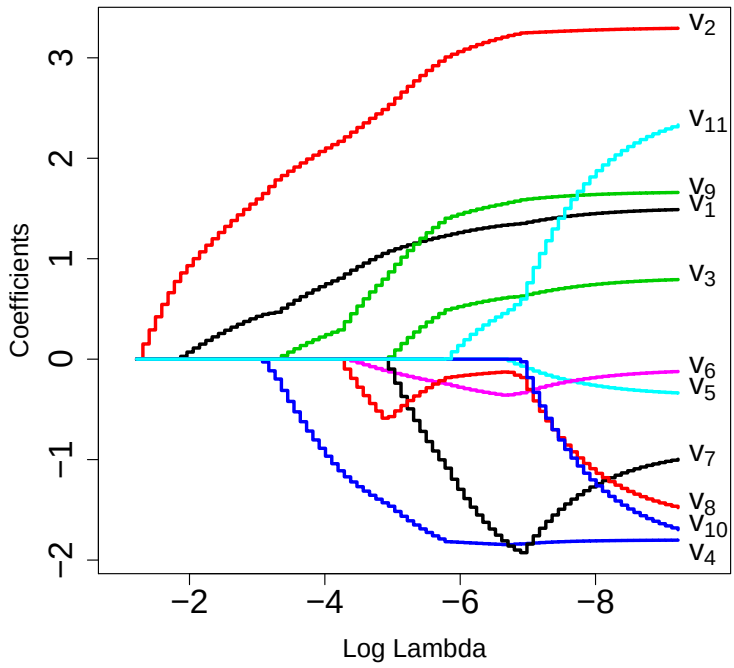


Figure 2.1: Example regularisation path produced using `glmnet`. Each coloured line corresponds to a different covariate. The plot describes how the value of the coefficient (y-axis) for each covariate changes as λ is varied (x-axis). Note that each of the coloured lines appears as a series of steps, highlighting the fact that only a *discrete* grid of λ values is considered.

shrinking the magnitude of the coefficients, but *without* setting any of them to be exactly zero. It follows that ridge regression does not by itself perform automatic feature selection. In contrast, the *mixture* of ridge and lasso penalties that defines the elastic net retains this property of the lasso (for $\alpha \in (0, 1]$). Moreover, for $\alpha \in [0, 1)$, the elastic net exhibits the *grouping effect* (Zou and Hastie, 2005), meaning that the estimated coefficients of strongly correlated covariates will tend to have similar values. In the extreme case where there are two or more perfectly collinear covariates, the corresponding coefficient values will be identical. The lasso does not have this property. Indeed, if there is a group of several strongly correlated covariates, then the lasso will tend to select just one of them (Zou and Hastie, 2005). The lasso is therefore "efficient" in its selection of covariates, in that it automatically strips out "redundant" covariates that provide little further information over those that have already been selected. As we now discuss, however, this property provides difficulties for stability selection.

2.4 Stability selection

The estimated selection probabilities of Section 2.2.5 provide us with a "stability score" for each of the covariates. Meinshausen and Bühlmann (2010) propose to use this score in order to pick out only the most stably selected covariates. Below, we provide a brief overview of the ideas behind this *stability selection* approach, after which we introduce *stability paths*, and then discuss in Section 2.4.2 some of the problems that occur due to correlations amongst the covariates,

Following Meinshausen and Bühlmann (2010), the set of stably selected covariates is defined to be,

$$S^{\text{stable}} = \left\{ v_i : \left(\max_{\lambda \in \Lambda} \left(\widehat{\text{P}}_{\text{Sel}}(\{v_i\}|\lambda, \mathcal{H}) \right) \right) \geq \pi_{\text{thr}} \right\}, \tag{2.12}$$

where Λ is a range of possible values for the regularisation parameter, λ; $\widehat{\text{P}}_{\text{Sel}}(\{v_i\}|\lambda, \mathcal{H})$ is the estimated marginal selection probability for covariate v_i; and π_{thr} is a probability threshold. In other words, S^{stable} is simply the set of covariates whose estimated marginal selection probability, $\widehat{\text{P}}_{\text{Sel}}(\{v_i\}|\lambda, \mathcal{H})$, is greater than a predefined threshold value (for any $\lambda \in \Lambda$).

Assuming that the covariates, v_i, may be considered as either "relevant" or "noise" variables (where relevant variables are those covariates which have a causal relationship with the output, y, and conversely for noise variables), and under a few further technical assumptions, a bound may be established on the expected number of noise variables appearing in S^{stable} (we refer to Meinshausen and Bühlmann, 2010, for full details).

Meinshausen and Bühlmann (2010) propose to control this bound through the selection of the threshold, π_{thr}, and the set, Λ. The resulting set of stably selected covariates, S^{stable} is then returned as the final set of selections. Meinshausen and Bühlmann (2010) demonstrate this procedure using both linear (in the regression case) and logistic (in the binary classification case) regression models, and define $\mathcal{H}$ by using the lasso to perform automatic feature selection.

2.4.1 Stability paths

One very useful way to visualise the stability selection procedure is provided by so-called *stability paths* (Meinshausen and Bühlmann, 2010). For each covariate, v_i, these describe how the estimated probability of selection, $\widehat{\text{P}}_{\text{Sel}}(\{v_i\}|\lambda, \mathcal{H})$, varies as a function of λ. They are therefore identical to conventional regularisation paths (Figure 2.1), except that the y-axis describes the estimated probability of selection associated with each covariate, rather than the estimated coefficient value. An example (which we discuss in more detail in the next section) is provided by Figure 2.2a . Note that the x-axis in this plot corresponds to a normalised version of the regularisation parameter, $\lambda_{\text{norm}} = \lambda/\lambda_{\max}$, so that $\lambda_{\text{norm}} = 1$ corresponds to $\lambda = \lambda_{\max}$.

2.4.2 The effects of correlation

Correlations amongst the "relevant" covariates can cause difficulties for stability selection. As noted in Kent (2010), if there are two strongly correlated relevant covariates, v_1 and v_2, then the lasso might select v_1 approximately half time, and v_2 approximately half the time (and rarely ever select both together). The effect of this would be to make both covariates appear relatively unstable, and might result in neither being selected (depending on the value of π_{thr} in Equation 2.12). This problem was considered further in Kirk *et al.* (2010), and the analysis is repeated below.

2.4.2.1 Simulation model

We consider a simulated regression example, in which $p = 500, n = 50$, the covariates are sampled from a $\mathcal{N}(0, \Sigma)$ distribution, and the response is given by $Y = \sum_{i=1,\ldots,8} v_i + \epsilon$, where ϵ is a zero-centred Gaussian noise term with variance 0.1. Here, Σ is the identity matrix except for the elements $\Sigma_{1,2} = \Sigma_{3,4} = \Sigma_{4,5} = \Sigma_{3,5} = 0.8$ and their symmetrical counterparts. It follows that there are only 8 "relevant" variables, $v_1, \ldots, v_8$, and that, amongst these, there are two strongly correlated sets: $\{v_1, v_2\}$ and $\{v_3, v_4, v_5\}$.

2.4.2.2 Probability sharing amongst correlated covariates

In order to assess the effects of correlation upon stability, we simulate 1,000 times from the model, and — for each simulated data set — use a subsampling strategy identical to the one employed by Meinshausen and Bühlmann (2010) (i.e. subsampling 50% of the data 100 times) to estimate the selection probabilities, $\widehat{\text{P}}_{\text{Sel}}(\{v_i\}|\lambda, \mathcal{H})$. The selection strategy, $\mathcal{H}$, is again defined to be automatic feature selection using the lasso. In Figure 2.2a, we show the stability path for one of these simulations. Note that v_2 appears stably selected, while the estimated selection probability for v_1 remains low for all values of λ. Indeed, the estimated probability of selection for v_1 is comparable to the estimated selection probabilities for the noise variables. For this particular simulation, we would therefore expect the stability selection procedure to fail to pick out v_1. This pattern is not universal across all simulations, however. For some simulations, it is v_1 that is stably selected, while the estimated selection probability for v_2 hovers around the noise level. For others, we observe the scenario envisaged by Kent (2010), with v_1 and v_2 both having estimated selection probabilities of around 0.5 or 0.6. In fact, there is a whole spectrum of behaviours, all of which follow a clear pattern: there is a negative relationship between the estimated selection probabilities for v_1 and v_2. This is illustrated in Figure 2.2b. Another way to think of this is to regard the selection probabilities as being "shared" amongst the correlated covariates (so, one selection probability might be high and one might be low, or both might be middling, but we will rarely see both being particularly high).

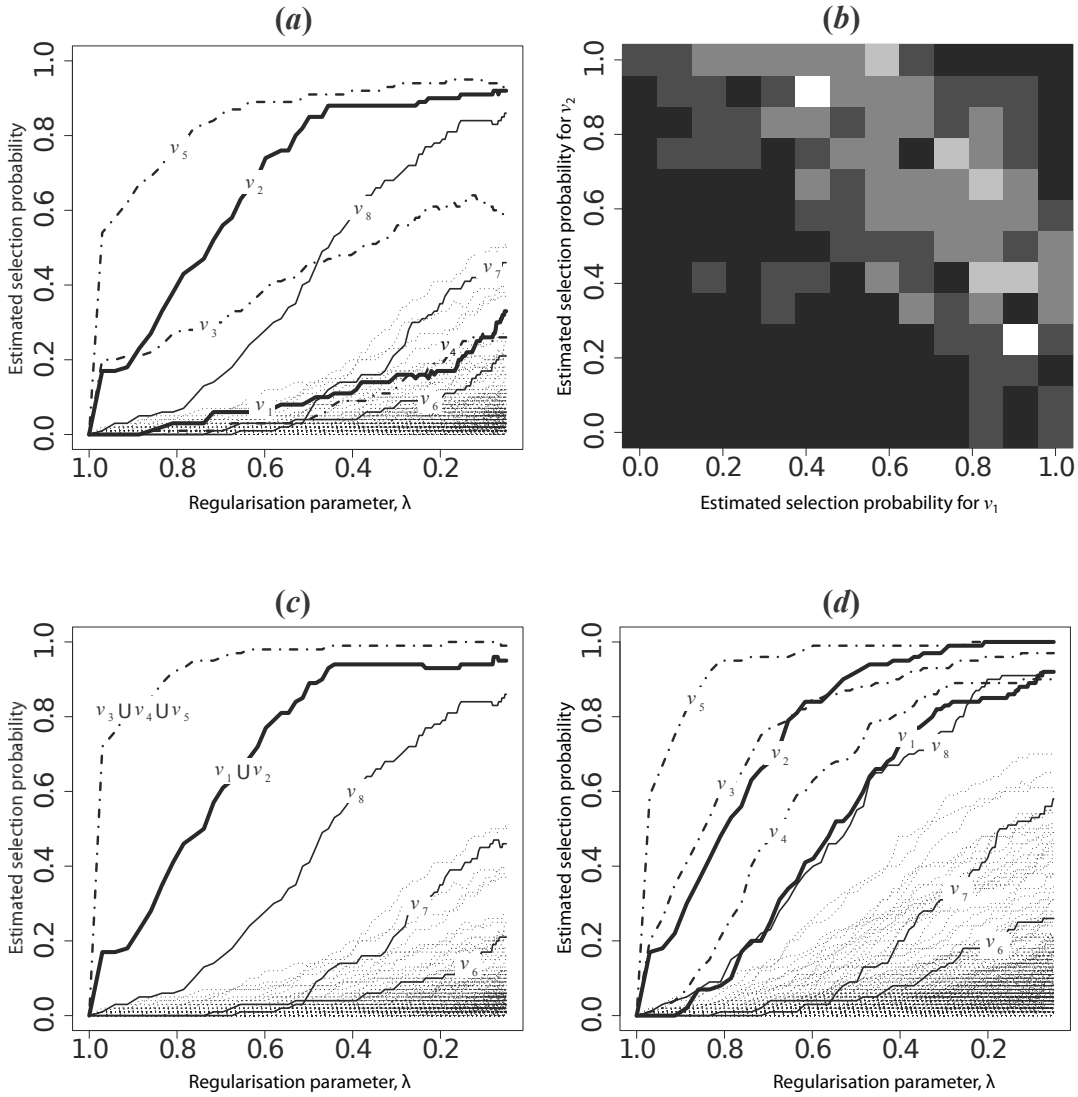


Figure 2.2: (a) Stability path for a particular realisation of the simulation example of Section 2.4.2. Thick black lines are used for v_1 and v_2, dot-dashed lines for v_3, v_4 and v_5, thin solid lines for v_6, v_7 and v_8, and faint, dotted lines for all noise variables; (b) For 1,000 realisations, selection probabilities for v_1 and v_2 at $\lambda = 0.25$ are estimated using a subsampling method. The plot illustrates the density of these points in the 0–1 square (with lighter squares indicating higher density), showing a clear negative relationship; (c) Using the same realisation as in Figure 2.2a, we plot the stability path when correlated covariates are grouped together; (d) Again using the same realisation, we obtain a stability path using the elastic net (with α set to 0.2).

2.4.2.3 Grouping correlated covariates

Figures 2.2a and 2.2b illustrate that, if the relevant variables are strongly correlated, stability selection (with the lasso) will often fail to select all of them. One way to overcome this is to perform a preprocessing step in order to detect the sets of correlated covariates. We may then consider the estimated probability of selecting at least one of the elements of the correlated set; for example, the estimated probability of selecting v_1 *or* v_2 (or both). If we adopt this approach, we obtain stability paths such as the one shown in Figure 2.2c.

2.4.2.4 Using the elastic net

Grouping the covariates together into correlated sets greatly alleviates the problems presented by having strong correlations amongst the relevant varables. However, in practice, defining these groups is likely to present further challenges. For example, how should we determine the critical level of correlation at which covariates should be grouped together? Assuming that we can overcome this problem, what should we do if we decide that covariates v_i and v_j should be grouped together, and similarly for v_i and v_k, but we then find that the correlation between v_j and v_k falls below our critical value?

A more elegant and straightforward approach would seem to be to use the elastic net (Section 2.3.3) in place of the lasso. Figure 2.2d demonstrates the improvements that such an approach can provide.

2.5 Selection by stability and predictive performance

The stability selection approach of Meinshausen and Bühlmann (2010) not only provides a useful method by which to assess selection stability, but also makes use of this information in order to define a new feature selection algorithm (albeit one that relies upon the "base" strategy, $\mathcal{H}$). There are, however, limitations to this approach (in addition to the problems with correlation that we identified in the previous section). For example:

1. The focus on bounding the number of falsely selected covariates means that, although the "false positive rate" (i.e. the frequency with which a noise variable appears in the stable set) is low, the "false negative rate" (i.e. the frequency with which a relevant variable is omitted) can be very high. The result of this is that the stability selection approach tends to be conservative — although we can say with high confidence that the covariates appearing in the stable set are relevant, we might be doing so at the expense of omitting other relevant covariates.

2. In practice, the clear distinction between "relevant" and "noise" covariates might be artificial. It might be more realistic to suppose that there is a continuum of

relevance, and that our aim is simply to identify the covariates that provide us with the best ability to predict.

These two limitations motivate the work in this section, where we consider an alternative stability selection approach that takes predictive performance into account. In brief, our strategy is to use the data left out of each subsample in order to quantify the predictive performance afforded by the selected covariates. We then combine this information with an assessment of stability, and finally choose the set of covariates that optimises both stability and predictive performance. In contrast to Meinshausen and Bühlmann (2010), our approach is only applicable for classification problems.

2.5.1 Assumptions

For the time being, we allow $\mathcal{H}$ to represent any selection strategy, subject to the following two requirements:

1. We assume that $\mathcal{H}$ may be parameterised by the number, m, of selections that it makes. For example, we might specify $m = 3$, in order to ensure that our strategy will return 3 selections. This is partly for ease of exposition: for a more general regularisation parameter, λ, we cannot immediately tell whether, for example, $\lambda = 0.2$ corresponds to 10 selections, 53 selections, or any other number between 0 and p. Moreover, any given value for λ will generally correspond to different numbers of selections depending upon the particular data set we observe. So, for replicate data sets D_i^{rep} and D_j^{rep}, the sizes of the selected sets $s(D_i^{\text{rep}}; \lambda, \mathcal{H})$ and $s(D_j^{\text{rep}}; \lambda, \mathcal{H})$ may well be different, even for the same value for λ. This is not necessarily problematic, but we believe it to be more intuitive and interpretable if we deal directly with the number of selections, m, rather than some technical parameter, λ. This is particularly the case if we wish to specify in advance a maximum number of covariates to be returned by our selection strategy. For example, if we wish to identify putative biomarkers, available resources may determine a maximum number of proteins that we are able to validate experimentally or test clinically. In what follows, we emphasise this assumption by writing m in the place of λ.

2. Accompanying $\mathcal{H}$, we require there to be a corresponding classification model, h, that allows us to make predictions (in the case of *embedded* selection strategies, $\mathcal{H}$ and h may be one and the same). For example, if logistic regression with an elastic net likelihood penalty is employed in order to select a set of covariates, then an appropriate form for the corresponding classifier h is a logistic regression model. If only a subset of the covariates $V \subseteq \{v_1, \ldots, v_p\}$ appears in the predictive model h, then we make this clear by writing $h[V]$.

We note that the first requirement is not excessively limiting, as most selection strategies can be formulated in such a way as to generate a ranked list of covariates from which we may then select the top m (for example, see Section 2.6.1.1).

2.5.2 Quantifying stability

We employ the estimated selection probabilities of Section 2.2.5. However, in contrast to Meinshausen and Bühlmann (2010), we focus upon estimated selection probabilities for sets V such that $|V| = m$ (rather than the marginal selection probabilities of individual covariates). That is, we consider,

$$\widehat{\mathrm{P}}_{\mathrm{Sel}}(V|m, \mathcal{H}) = \frac{1}{B} \sum_{i=1}^{B} \mathbb{I}\left(V = s(D_i^{\mathrm{rep}}; m, \mathcal{H})\right), \tag{2.13}$$

where we now have a strict equality inside the indicator function. In words, $\widehat{\mathrm{P}}_{\mathrm{Sel}}(V|m, \mathcal{H})$ is simply the proportion of the B subsampled data sets for which the corresponding selection of size m was V.

2.5.3 Quantifying predictive performance

When we have only a limited number of observations, we often employ a cross-validation approach in order to assess predictive performance (see Section 1.4.2.1). In random subsample cross-validation, we proceed by first drawing subsamples $D_1^{\mathrm{rep}}, \ldots, D_B^{\mathrm{rep}}$ from D^{obs}. For each i we then train classifier h on D_i^{rep} and calculate the correct classification rate, $c(D_i^{\mathrm{rep}}; h)$, when h is applied to $D_{\setminus i}^{\mathrm{rep}} = D^{\mathrm{obs}} \setminus D_i^{\mathrm{rep}}$ (note that the correct classification rate is simply one minus the misclassification rate). Similar to the way in which Equation (2.13) is identified as an estimate of the probability of selecting set V, we may identify the mean of the B correct classification rates $c(D_i^{\mathrm{rep}}; h)$ as an estimate of the probability that h classifies correctly.

Returning to our problem, we wish to estimate the probability of correct classification using $h[V]$, given that V is one of our selected sets of size m. In line with the random subsample cross-validation procedure outlined above, and making use of the existing subsampling strategy used for the selection probabilities, we estimate this as,

$$\widehat{P}_{\mathrm{Correct}}(h[V]|V = s(D_i^{\mathrm{rep}}; m, \mathcal{H}) \text{ for some } i) := \frac{\sum_{i=1}^{B} c(D_i^{\mathrm{rep}}; m, \mathcal{H}) \mathbb{I}(s(D_i^{\mathrm{rep}}; m, \mathcal{H}) = V)}{\sum_{i=1}^{B} \mathbb{I}(s(D_i^{\mathrm{rep}}; m, \mathcal{H}) = V)}, \tag{2.14}$$

where $c(D_i^{\mathrm{rep}}; m, \mathcal{H})$ is the correct classification rate achieved when $h[s(D_i^{\mathrm{rep}}; m, \mathcal{H})]$ is trained on D_i^{rep} and applied to $D_{\setminus i}^{\mathrm{rep}}$.

Thus, $\widehat{P}_{\mathrm{Correct}}(h[V]|V = s(D_i^{\mathrm{rep}}; m, \mathcal{H}) \text{ for some } i)$ is simply the average of the correct classification rates achieved whenever V was selected.

2.5.4 Combining stability and predictive performance

Equation (2.14) provides an estimate of the conditional probability of correct classification using $h[V]$, given that V is a selected set. Equation (2.13) estimates the probability that V is selected. We may hence estimate the joint probability of both selecting V *and* classifying correctly using the resulting classifier $h[V]$ by multiplying these estimates together. We therefore obtain,

$$\widehat{P}_{\text{Joint}}(V|m, \mathcal{H}, h) := \frac{1}{B} \sum_{i=1}^{B} c(D_i^{\text{rep}}; m, \mathcal{H}) \mathbb{I}(s(D_i^{\text{rep}}; m, \mathcal{H}) = V). \tag{2.15}$$

Note that this is identical to Equation (2.14), except that rather than dividing by the number of subsamples for which V was selected, we divide by the *total* number of subsamples.

It is this joint probability of selection and correct classification (which, for brevity, we will henceforth refer to as the "joint score") that we shall use in order to assess the quality of a given set V, and which we will ultimately optimise in order to make our final selections. Note that Equation (2.15) is just a weighted estimate of the probability of correct classification (Equation 2.14). This weighting penalises sets that are unstably selected, since B will be large relative to $\sum_{i=1}^{B} \mathbb{I}(s(D_i^{\text{rep}}; m, \mathcal{H}) = V)$ for sets that are selected only a few times. This has the additional positive side-effect of down-weighting estimates that are made on the strength of small numbers of subsamples. If V is a completely stable set (i.e. for all D_i^{rep}, we have $s(D_i^{\text{rep}}; m, \mathcal{H}) = V$), then $\sum_{i=1}^{B} \mathbb{I}\left(s(D_i^{\text{rep}}; m, \mathcal{H}) = V\right) = B$, and hence Equations (2.14) and (2.15) coincide.

2.5.5 Selecting by joint score maximisation

The above discussion suggests a selection strategy in which we seek the set V that maximises Equation (2.15). Given a range of values for m (say, $m = 1, \ldots, m_{\max}$) and a collection of different selection algorithms $\{\mathcal{H}_k\}_{k=1}^{K}$ (and corresponding classification models $\{h_k\}_{k=1}^{K}$), we perform an exhaustive search in order to find,

$$\widehat{V}(m, k) := \underset{V}{\operatorname{argmax}} \left\{ \widehat{P}_{\text{Joint}}(V|m, \mathcal{H}_k, h_k) \right\}, \tag{2.16}$$

and

$$\widehat{P}(m, k) := \max \left\{ \widehat{P}_{\text{Joint}}(V|m, \mathcal{H}_k, h_k) \right\}, \tag{2.17}$$

for $1 \leq m \leq m_{\max}$ and $1 \leq k \leq K$. For each $\mathcal{H}_k$ we can then plot how the maximal value of the joint score varies with increasing m, which allows for straightforward comparisons of different selection strategies (as we shall see in Figure 2.5). As our final selected set and final choices for m and k we take,

$$\widehat{m}, \widehat{k}, \widehat{V} := \underset{m,k,V}{\operatorname{argmax}} \left\{ \widehat{P}_{\text{Joint}}(V|m, \mathcal{H}_k, h_k) \right\}. \tag{2.18}$$

For complete clarity, a straightforward algorithm for computing these quantities is presented in Algorithm 1.

Algorithm 1 Selection by stability and predictive performance

Draw subsamples $D_1^{\text{rep}}, \ldots, D_B^{\text{rep}}$ from D^{obs}
topScore $\leftarrow 0$
topSelectedSet $\leftarrow$ emptySet()
for $k = 1$ to K **do**
  $\mathcal{H} \leftarrow \mathcal{H}_k$
  $h \leftarrow h_k$
  for $m = 1$ to $m_{\max}$ **do**
    selectedSets $\leftarrow$ emptyList()
    classRates $\leftarrow$ emptyVector()
    counter $\leftarrow 1$
    for $i = 1$ to B **do**
      $V \leftarrow s(D_i^{\text{rep}}; m, \mathcal{H})$
      if $V \in$ selectedSets **then**
        index $\leftarrow$ which(selectedSets == V)
        classRates[index] $\leftarrow$ classRates[index] $+c(D_i^{\text{rep}}; m, \mathcal{H})$
      else
        selectedSets[counter] $\leftarrow V$
        classRates[counter] $\leftarrow c(D_i^{\text{rep}}; m, \mathcal{H})$
        counter $\leftarrow$ counter $+1$
      end if
    end for
    if max(classRates) $\geq$ topScore **then**
      index $\leftarrow$ which(classRates == max(classRates))
      topScore $\leftarrow$ max(classRates)
      topSelectedSet $\leftarrow$ selectedSets[index]
    end if
  end for
end for

It is clear from Algorithm 1 that the procedure for finding the set that optimises the joint score may be computationally costly, since we have to perform feature selection, train our classification model, and then evaluate its predictive performance a total of $KBm_{\max}$ times. We note, however, that this part of the procedure is eminently parallelisable: we may consider each value of m for each $\mathcal{H}_k$ applied to each D_i^{rep} on a separate computer node. We must simply ensure that each node returns a selected set, V, and a corresponding correct classification rate. Subsequently, we must process these outputs (together) by identifying all of the distinct V's amongst those that were returned, and — for each of these — we must sum all of the corresponding classification rates. The V that has the highest sum is the final output of the algorithm.

2.6 Implementation

Until now, our exposition regarding how to select by stability and predictive performance has been rather general. We now provide details of how the subsampling is performed, as well as describing the selection strategies and classification model that we consider.

2.6.1 Subsampling

Suppose that n_0 of the observations in D^{obs} belong to class 0, and $n_1 = n - n_0$ belong to class 1. Then, in all of the examples considered in this paper, we form subsamples $D_1^{\text{rep}}, \ldots, D_B^{\text{rep}}$ which maintain the class proportions of the original set by each time randomly sampling $0.5\lfloor n_j \rfloor$ observations from class j, for $j \in \{0, 1\}$. For the simulation example of Section 2.7, we take the number of subsampled data sets to be $B = 100$; and for the biomarker discovery example of Chapter 3 we take $B = 250$.

2.6.1.1 Selection strategy

As previously, we focus on selection strategies that use logistic regression models with elastic net likelihood penalties, including the lasso penalty as a special case. In the following, we consider a grid of α values ($\alpha = 0.1, 0.2, \ldots, 1$), and find for each the *regularisation path* (Figure 2.1). Each different value of α defines a different covariate selection strategy, so that we have 10 strategies $\mathcal{H}_1, \ldots, \mathcal{H}_{10}$, with H_k corresponding to $\alpha = k/10$. Throughout, we use the `glmnet` package in `R` (Friedman *et al.*, 2010) in order to construct regularisation paths. We form a ranked list from the regularisation path by considering the order in which covariates are selected (i.e. the order in which coefficients become non-zero as we decrease λ). If, due to the effects of the regularisation path being evaluated on a discrete grid of λ values, two covariates seem to appear simultaneously (e.g. covariates v_3 and v_7 in Figure 2.1), then we randomly choose which one should appear first in the list. In principle, this random choice errs on the side of slightly decreasing the stability of our selections; in practice, however, we find it to have little effect. The selected set of size m is then defined to be the top m covariates in the ranked list.

2.6.2 Predictive model

For each selection strategy, $\mathcal{H}_k$, we take h (the corresponding predictive model) to be a logistic regression classifier. We train h by unpenalised maximisation of the log-likelihood. We note that this two-step procedure of using the elastic net for variable selection and then obtaining unpenalised estimates of the coefficients in the predictive model is similar to the LARS-OLS hybrid (Efron *et al.*, 2004) or the relaxed lasso (Meinshausen, 2007).

2.6.3 The `stabSel` package

We have implemented the procedure described in this section as part of an `R` package, which we call the `stabSel` package. This additionally allows the calculation of estimated marginal selection probabilities, and the Jaccard and Kuncheva similarity coefficients. The user may vary the number of subsamples, subsample proportions etc., or may use predefined defaults.

2.7 Simulation example

We consider a similar example to the one presented in Section 2.4.2.1. We have $p = 500$ predictors $v_1, \ldots, v_{500}$ and $n = 200$ observations. The predictors $v_1, \ldots, v_{500}$ are jointly distributed according to a multivariate normal whose mean μ is the zero vector and whose covariance matrix Σ is the identity, except that the elements $\Sigma_{1,2} = \Sigma_{3,4} = \Sigma_{3,5} = \Sigma_{4,5}$ and their symmetric counterparts are equal to 0.9. Thus, there are two strongly correlated sets, $C_1 = \{v_1, v_2\}$ and $C_2 = \{v_3, v_4, v_5\}$, but otherwise the predictors are uncorrelated. Observed class labels y are either 0 or 1, according to the following logistic regression model:

$$P(y = 1|v_1, \ldots, v_{500}) = \frac{1}{1 + \exp\left(-\sum_{i=1}^{5} v_i\right)}. \tag{2.19}$$

Since $v_1, \ldots, v_5$ are the only covariates that appear in the generative model given in Equation (2.19), the notion of "relevant" and "noise" variables is appropriate here.

We simulate 1,000 data sets by first sampling from a multivariate normal in order to obtain realisations of the covariates $v_1, \ldots, v_{500}$, and then generating values for the response y according to Equation (2.19). When performing selections, $m_{\max}$ is set to 20.

2.7.1 Selecting by predictive performance

In order to assess the usefulness of our approach, we compare it to a selection strategy based upon predictive performance alone. We proceed as before, taking subsamples $D_1^{\text{rep}}, \ldots, D_B^{\text{rep}}$ of data set D^{obs} (where we use the same proportions and value of B as for the joint selection strategy). For a given value of m, we apply strategy $\mathcal{H}$ to subsampled data set D_i^{rep} in order to obtain a selected set $V = s(D_i^{\text{rep}}; m, \mathcal{H})$. We then train the predictive model $h[V]$ on D_i^{rep} and assess its performance on the left out set $D_{\setminus i}^{\text{rep}}$. We hence obtain a correct classification rate $c(D_i^{\text{rep}}; m, \mathcal{H})$. By taking the average over all subsamples, we obtain the mean correct classification rate, $\overline{c}(m, \mathcal{H}) = \sum_{i=1}^{B} c(D_i^{\text{rep}}; m, \mathcal{H})/B$. Given a range of values of m and several different models $\mathcal{H}$, we select the combination $(m_{\text{sel}}, \mathcal{H}_{\text{sel}})$ which maximises $\overline{c}(m, \mathcal{H})$. The final selected set, $s(D; m_{\text{sel}}, \mathcal{H}_{\text{sel}})$, is given by applying $\mathcal{H}_{\text{sel}}$ to the original data set D^{obs} in order to obtain m_{sel} selections.

2.8 Results

We present in this section the results obtained from our simulation example, before moving on to a real biological example in the next chapter.

2.8.1 No false positives

We applied our selection strategy (Algorithm 1) to each of our 1,000 simulated data sets. For 50.5% of simulations, we selected all 5 relevant covariates; in a further 39.1%, we selected just covariates from the second correlated set; and in the remainder we selected various combinations of the 5 relevant covariates (Figure 2.3, left bar). Our strategy never selected a set containing a noise covariate. This is in stark contrast to the selection strategy based upon predictive performance alone (Section 2.7.1), as illustrated in Figure 2.3, right bar. For 26.4% of our simulated data sets, we obtained selected sets that contained at least one noise variable. Not only is this "false positive" rate significantly higher than for our "joint score" selection strategy, but also the "true positive" rate is lower, with the full set of relevant covariates being selected for only 26.3% of the simulated data sets .

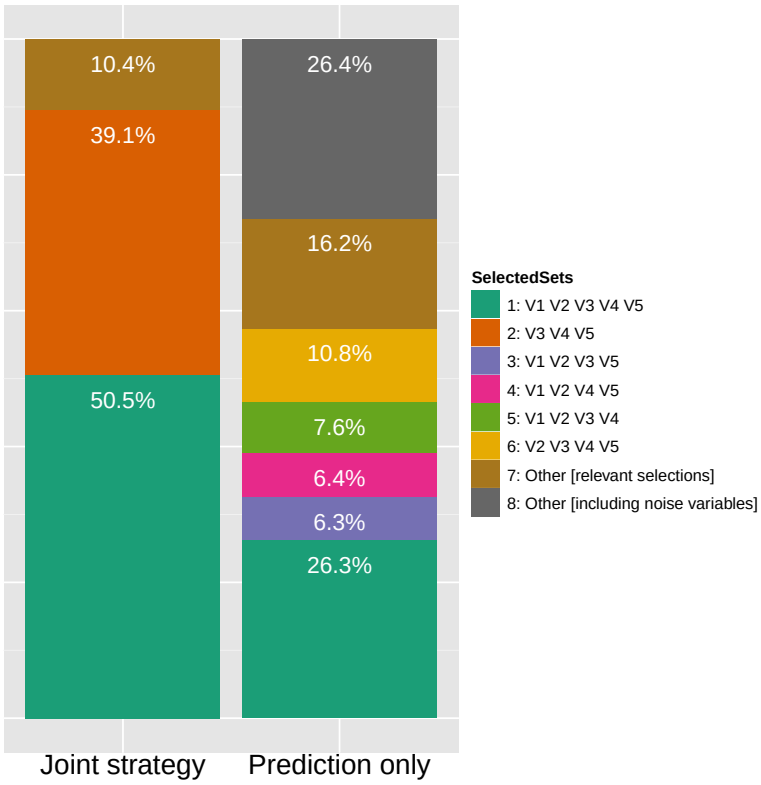


Figure 2.3: Sets selected using "joint score" and "predictive performance" methods.

2.8.2 Smaller values of α yield higher scoring selections

Figure 2.4 illustrates the sampling distributions (over all 1,000 simulations) of the maximal joint scores (Equation 2.17) for different values of α. We can see that smaller values of α tended to yield higher maximal values of the joint score.

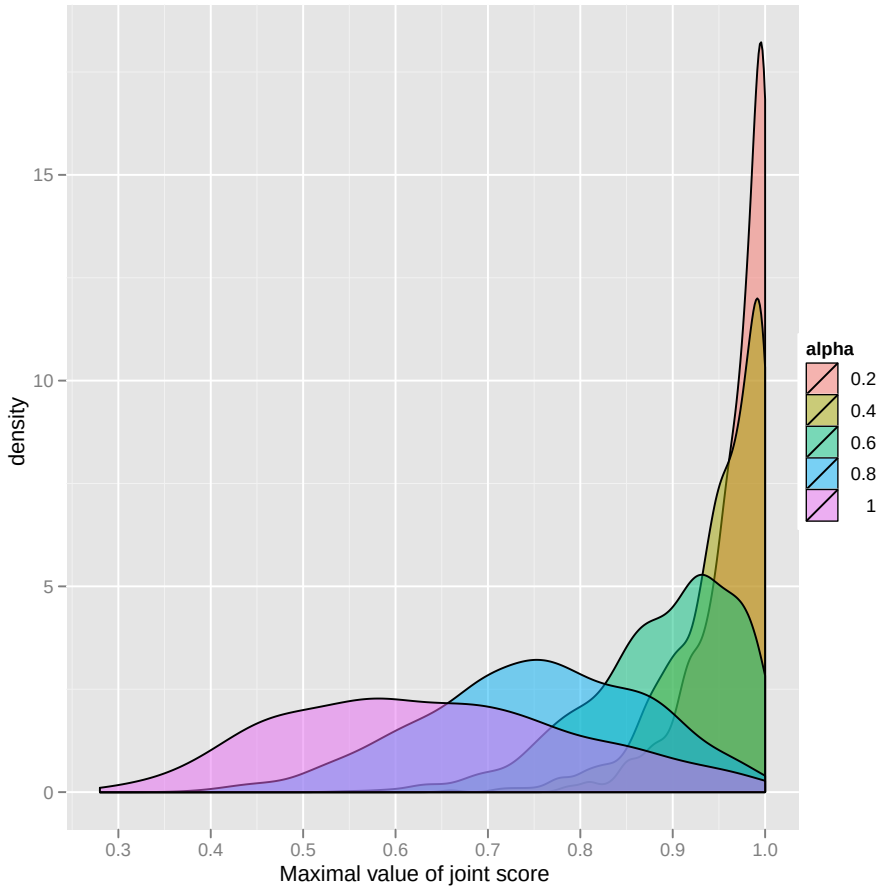


Figure 2.4: Distributions of the maximal joint score values for 5 different values of α.

2.8.3 Analysis of a single simulation

As well as looking at the average performance over all 1,000 simulations, it is also useful to consider a single simulation, as this corresponds to the more realistic scenario in which we have only one data set. In this section, we illustrate results for one particular simulation, which we refer to as "Simulation 1". In Figure 2.5, we show how the maximal joint scores (Equation 2.17) vary as a function of m and α. Amongst all considered values of m and α, the absolute maximum was achieved when $m = 5, \alpha = 0.2$ and $V = \{v_1, v_2, v_3, v_4, v_5\}$ (as indicated in the figure). This set is therefore returned by Algorithm 1 as our final selection.

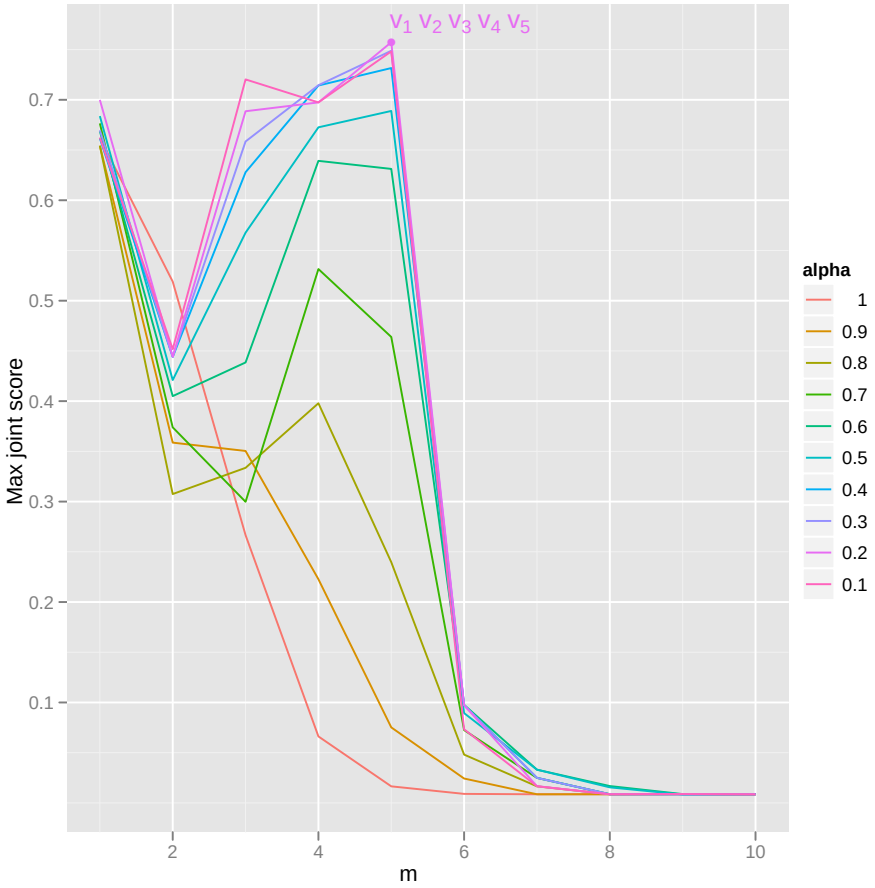


Figure 2.5: Maximal joint scores as a function of m for Simulation 1. We show the results from all 10 values of α that we considered.

2.8.3.1 Comparison to Jaccard and Kuncheva similarity indices

We may additionally consider the Jaccard and Kuncheva indices as alternative ways in which to quantify stability. For each value of m (and each value of α), we have a collection of 100 selected sets, $s(D_i^{\text{rep}}; m, \mathcal{H})$, of size m. We may hence calculate the mean pairwise Jaccard and Kuncheva coefficients for these collections (Equations 2.2.2 and 2.2.3). If we plot the resulting values as a function of m, we obtain *similarity paths*, as shown in Figure 2.6.

We note that there is a high degree of similarity between Figure 2.5 and Figures 2.6a and 2.6b. This is reassuring, and suggests that aspects of our approach could perhaps be employed in existing methodologies that currently utilise the Jaccard or Kuncheva similarity indices (such as Abeel *et al.*, 2010). The joint score has a number of advantages over these (and other) similarity measures. Principal amongst these is that the joint score has a straightforward probabilistic interpretation (while the similarity measures tend to be more heuristic in nature), and also that it incorporates an assessment of predictive ability.

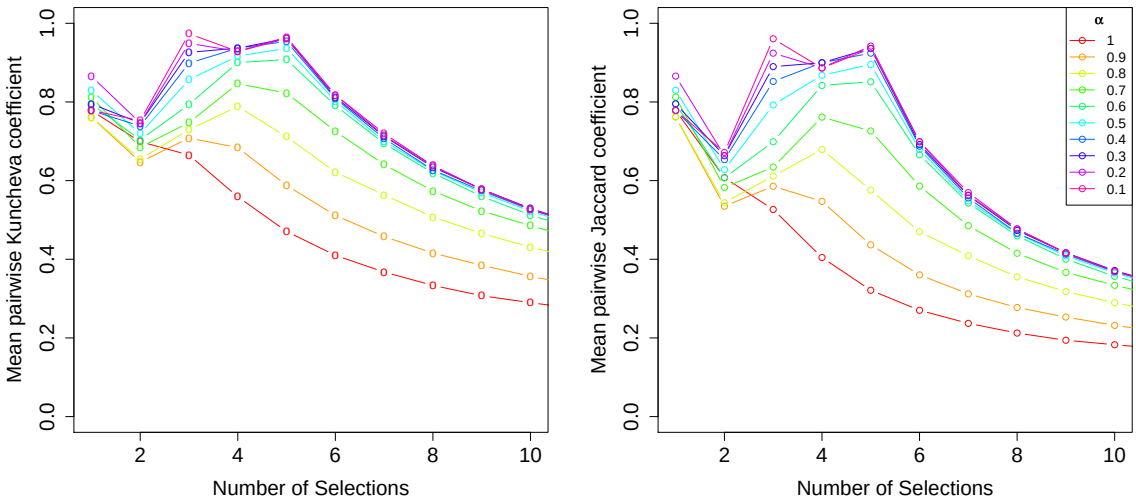


Figure 2.6: (a) Kuncheva, and (b) Jaccard similarity paths for different values of α (for Simulation 1).

2.9 Discussion

We have considered a number of ways for assessing the stability of covariate selections. We discussed the recently published *stability selection* procedure of Meinshausen and Bühlmann (2010), and highlighted the difficulties that can arise as a result of correlations amongst the covariates. We then presented a novel score for combining assessments of stability and predictive performance. For a subset V of the covariates (and given selection strategy $\mathcal{H}$ and classification model h), this score may be interpreted as an estimate of the joint probability of selection of V by $\mathcal{H}$ and correct classification using $h[V]$. We further constructed a straightforward algorithm which returns the set V (as well as the number of selections, m, and model indicator, k) that maximises our score. This algorithm is implemented in the `stabSel` package in `R`. The algorithm allows different covariate selection strategies and classification models to be considered together, and also permits us to compare their stability and predictive properties. We employed a selection strategy using logistic regression models with the elastic net likelihood penalty. By considering a range of values for the α parameter, we were able to investigate the effects of correlation on stability.

We applied our algorithm to a series of simulated data sets for which we knew the "correct" covariate selections. For all 1,000 of our simulations, the sets returned by our algorithm included only relevant covariates. For just over half of the simulations we recovered *all* of the relevant covariates. This represented a significant improvement in performance relative to a selection strategy based upon predictive performance alone, which returned all of the relevant covariates for only a quarter of the simulated data sets, and which also frequently selected noise covariates. Looking more closely at our results enabled us to determine that the covariate selections with the highest joint scores tended to be given

by lower values of α. Given that there was a strong correlation structure amongst our covariates (and in light of the discussion of Section 2.4.2), this is unsurprising.

We then focused our attention upon a single simulation in order to replicate the situation in which we have just one data set. We demonstrated that the results obtained using our method were in good agreement with assessments of stability that employed the Jaccard or Kuncheva similarity indices.

Overall, we have demonstrated that our approach provides a useful way to combine assessments of stability and predictive performance, which has favourable properties relative to methods that select on the basis of predictive performance alone. Our algorithm allows us to compare several different selection strategies, $\mathcal{H}_k$. Its output tells us not only the number of selections and selected set which optimise the joint score, but also the optimal selection strategy from amongst those we considered (Equation (2.18)). In the next chapter, we apply our algorithm in the context of HTLV-1 biomarker discovery.

Chapter 3

HTLV-1 biomarker discovery

Abstract *We apply the algorithm of the previous chapter to SELDI-TOF MS blood serum data obtained from HTLV-1 seropositive patients. We identify a number of putative protein peak biomarkers to distinguish between asymptomatic carriers of the virus and infected individuals who suffer from an inflammatory condition known as HAM/TSP. Two of the putative biomarkers have been experimentally identified as β_2-microglobulin and Calgranulin B. Our results indicate that, together, these proteins allow us to discriminate between the two classes of HTLV-1 infected individuals with (approximately) a 78% cross-validation rate of correct classification.*

Outline *In Section 3.1 we provide information about both HTLV-1 and SELDI-TOF MS. We describe the data in Section 3.2, and perform some standard statistical analyses in Section 3.3. In Section 3.4 we apply the algorithm of Section 2.5. As part of our analysis, we demonstrate the difficulties caused by correlation that we identified in the previous chapter. We then discuss the plausibility of the putative protein peak biomarkers, with reference to further experimental results (Section 3.5). We provide further general conclusions in Section 3.6.*

3.1 Background

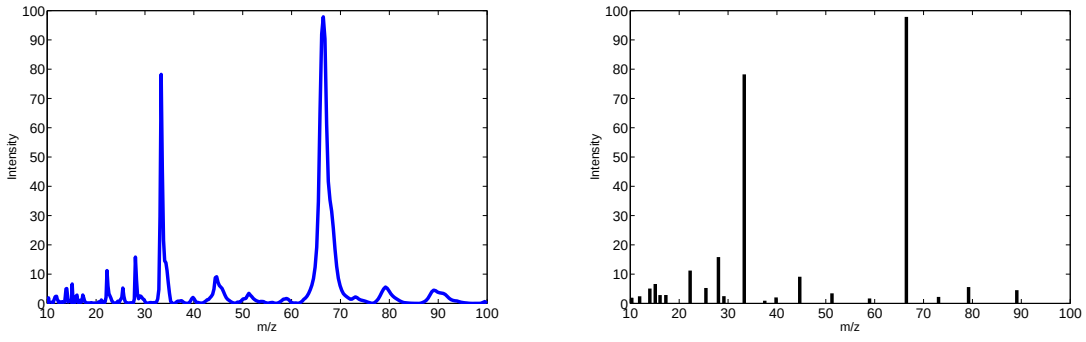
The human T lymphotropic virus Type 1 (HTLV-1) is a widespread virus associated with a range of diseases, the most commonly recognised of which are adult T-cell leukaemia/-lymphoma (ATLL) and an inflammatory condition of the central nervous system known as HTLV-1 associated myelopathy/tropical spastic paraparesis (HAM/TSP). In the present document, it is with the latter that we are concerned. HTLV-1 persists lifelong in the host, but only between 0.1% and 2% of infected individuals suffer from HAM/TSP (Bangham, 2000). It is therefore of significant interest to determine any factors that are different

between asymptomatic carriers (ACs) of the virus, and individuals with HAM/TSP. Previous research has focused upon characterising the lymphocyte population in HAM/TSP patients versus ACs (Goon *et al.*, 2003; Toulza *et al.*, 2008), comparing the expression levels of HTLV-1 genes (Asquith *et al.*, 2005; Toulza *et al.*, 2008), and investigating the integration sites in the host cell genome of the HTLV-1 provirus (Meekings *et al.*, 2008).

We here add to the emerging picture of the pathogenesis of HAM/TSP by considering a novel data-type; namely, surface-enhanced laser desorption/ionisation time of flight mass spectrometry (SELDI-TOF MS) proteomics data. We seek to identify proteins in the blood plasma whose abundances are associated with HAM/TSP, and which allow HAM/TSP patients to be discriminated from ACs.

3.1.1 SELDI-TOF MS

For brevity, we do not provide a detailed review of the technology used for SELDI-TOF MS, and refer to (Issaq *et al.*, 2002) for further information. To summarise, pre-prepared samples are spotted onto an array (chip), and are then analysed by a laser desorption/ionisation time of flight mass spectrometer. This produces a spectrum, such as the one shown in Figure 3.1a, which displays the mass/charge (m/z) ratio of the ionised proteins, and the signal intensity of the ions. Once a collection of such spectra has been generated, peak detection and clustering may be performed in order to reduce the dimensionality (and increase the interpretability) of each spectrum, producing a series of "spikes", such as those shown in Figure 3.1b. Each spike is assumed to correspond to a protein, with the intensity (height) being a measure of abundance.



(a) (b)

Figure 3.1: A section of a SELDI-TOF MS output. (a) Displayed as a spectrum. (b) Displayed as a series of "spikes" (after processing). The spikes summarise the key information for each peak: namely, it's mass/charge ratio (x-axis) and its intensity (y-axis).

Ideally, we might hope that we could simply "look up" the proteins corresponding to the mass/charge ratio for each spike, and hence easily identify the proteins in our samples. Unfortunately, the relative imprecision with which the technology assigns protein peaks

to molecular masses necessitates the use of additional experimental methodologies in order to perform the identification (Ndao *et al.*, 2010). It was therefore originally envisaged that the principal usage of SELDI-TOF MS data would be as part of a more complete data analysis (Issaq *et al.*, 2002). Candidate protein peak biomarkers would be identified from the original spectra, and then later identified by other means. However, later studies questioned the necessity of the protein identification step, arguing that — for diagnostic purposes — this information was not truly necessary, and that it was instead sufficient to determine "proteomic patterns" that allowed different types of samples (e.g. case/control) to be discriminated from one another (Petricoin *et al.*, 2002b). One of the results of this is that many subsequent studies worked with the full, high-dimensional spectral representation of the data, rather than the lower-dimensional, clustered peaks (Petricoin *et al.*, 2002a; Sorace and Zhan, 2003).

In a reanalysis of the data of Petricoin *et al.* (2002b), Baggerly *et al.* (2004) highlighted serious concerns regarding the reproducibility of results obtained from SELDI-TOF MS data, and identified a number of problems with the experimental protocols and statistical analyses that had been used in earlier papers. The identified issues included: an inability to generalise results to new data sets; being able to achieve perfect classification using "noise" in the spectra; and determining that, in Petricoin *et al.* (2002b), a "baseline correction" had not been performed[1] As a result of these issues, there has been a degree of distrust regarding results obtained from SELDI-TOF MS data (Constans, 2006). However, many authors have taken heed of the warnings and recommendations of Baggerly *et al.* (2004), with successful results (MacGregor *et al.*, 2008; Hand, 2008; Pusztai *et al.*, 2004). Current opinions of SELDI-TOF MS are much improved (Ndao *et al.*, 2010), particularly when viewed as a means for proposing putative biomarkers that are to be followed up in further experimental studies (MacGregor *et al.*, 2008).

In order to mitigate the issues raised by Baggerly *et al.* (2004), a carefully designed experimental study was undertaken, in which two separate data sets were generated (see Section 3.2). Appropriate preprocessing and normalisation of the data was performed, after which we applied a number of standard approaches for analysis (Section 3.3). Finally, in Section 3.4, we apply the selection algorithm of Section 2.5. Given the concerns about reproducibility of results, we believe it to be particularly appropriate to apply a stability selection approach to SELDI-TOF MS data such as these.

3.2 The data

Blood plasma samples were obtained from 68 HTLV-1-seropositive patients, 34 of whom were sufferers of HAM/TSP and 34 of whom were ACs. The data were collected in two batches, both of which comprised 17 HAM/TSP samples (henceforth HAMs) and

[1] Baseline correction refers to a procedure by which the baseline of the spectrum is "flattened" to a constant (zero) level in order to allow different spectra to be compared to one another (see Sauve and Speed, 2004).

17 ACs. Additionally, samples were obtained from 16 ethnically matched uninfected controls (henceforth U). Two data sets were formed, D_0 and D_V. The former comprises the first batch of HTLV-1 samples, plus all 16 Us, while the latter comprises the second batch of HTLV-1 samples, plus 14 of the Us.

The D_O samples were spotted onto a number of chips for SELDI-TOF MS analysis. A randomisation process was employed, which ensured that each chip included a variety of HAM, AC and U samples. Onto each chip, an independent control sample was also spotted (identical for all chips), so that we could later correct for any chip-specific effects (see Section 3.2.1). SELDI-TOF MS was used to generate spectra, and baseline correction and normalisation by total ion current (TIC) were performed using proprietary software (Biomarker Wizard, BioRad). The same software was used in order to detect and cluster protein peaks[2], the final output being a series of spikes. The samples from the D_V set were processed in an identical manner, but separately from the D_O set.

In addition to these two data sets, we also considered all 68 HTLV-1 samples together to create a "combined" data set, D_C. Peak detection and clustering were again performed using Biomarker Wizard, but this time with all 68 HTLV-1 spectra together.

3.2.1 Normalisation for chip-specific effects

Since the samples were distributed over several chips, we applied a normalisation procedure to control for any chip-specific effects. Given that the control samples spotted onto each chip are identical, we know that any discrepancy between them must arise as a result of chip-specific biases (and random noise). We therefore modelled the intensities of the peaks in the control spectra using a linear model of the type often used in analyses of microarray data (Churchill, 2004),

$$y_{ik} = A_i + B_k + \epsilon_{ik}. \tag{3.1}$$

Here, y_{ik} is the measured intensity of peak i in the spectrum from the control sample on chip k; A_i is the true unobserved intensity of peak i in the control sample (independent of the chip); B_k is a chip-specific fixed effect; and ϵ_{ik} is a zero-mean random error term. The moment estimator of the chip effect is then,

$$\hat{B}_k = \bar{y}_{\cdot k} - \bar{y}_{\cdot\cdot}, \tag{3.2}$$

where $\bar{y}_{\cdot k}$ is the average peak intensity across all of the peaks that appear in the control sample on chip k, and $\bar{y}_{\cdot\cdot}$ is the average peak intensity across all peaks in all control samples (i.e. across all chips).

[2]Clustering peaks refers to the procedure that is used in order to match up peaks from different spectra. For example, in one spectrum, there might be a peak with m/z value 13,301.7 and in a second spectrum, there might be a peak with m/z value 13,303.0. The clustering procedure analyses all spectra together and determines whether or not these peaks should be grouped together (and hence whether or not we should regard these peaks as corresponding to the same protein). See Tibshirani *et al.* (2004); Coombes *et al.* (2003) for typical examples of peak clustering algorithms.

We normalised all other spectra by subtracting the appropriate chip effect from the intensities. That is, we calculated,

$$y_{ij}^{\text{normalised}} = y_{ij} - \hat{B}_{k(j)}, \tag{3.3}$$

where y_{ij} is the (original) intensity for peak i in sample j, and $k(j)$ denotes the chip on which sample j was run.

3.2.2 Summary

To recap, after performing the SELDI-TOF MS analysis and the various preprocessing and normalisation steps, we have 3 data sets:

- D_O, comprising 17 HAMs, 17 ACs, 16 Us.
- D_V, comprising 17 HAMs, 17 ACs, 14 Us.
- D_C, comprising 34 HAMs, 34 ACs.

3.3 Preliminary data analysis

We start with a standard statistical analysis of the D_O and D_V data sets, in which we use a nonparametric alternative to the conventional t-test in order to identify protein peaks whose intensities differ significantly between HAMs and ACs. More precisely, we use the Mann-Whitney U test (Mann and Whitney, 1947), which we may view as testing for a significant difference between the median intensities of the two groups. We employ the variant of the Benjamini-Hochberg multiple testing correction procedure suggested by Storey (2002), controlling the false discovery rate at $q = 0.05$ (see also Section 1.2.2). As well as comparing the HAM and AC classes, we additionally consider the following pairwise comparisons: HAM vs. U, and AC vs. U.

The peaks determined to be significant are shown in Tables 3.1 and 3.2 for the D_O and D_V data sets respectively. Note that the peak identifiers are the mass-to-charge "locations" of the peaks, the units of which are kiloDaltons (kDa).

Tables 3.1 and 3.2 show that, for both the D_O and D_V data sets, the 11.7kDa, 11.9kDa, 13.3kDa and 14.6kDa peaks have significantly different median intensities between the HAM and AC classes (and also between the HAM and U classes). In Figures 3.2 and 3.3 we show heatmaps which illustrate the pattern of intensities for these peaks amongst the HAM and AC spectra. The figures show a clear difference between the intensities of these peaks in the two classes of samples: they are all elevated amongst the patients with HAM/TSP.

Our inability to detect significant differences between the median peak intensities of the AC and U classes is somewhat disappointing, but perhaps unsurprising. Although infected with HTLV-1, the ACs are, after all, asymptomatic, and hence we would expect to find it difficult to distinguish them from the uninfected controls.

HAM vs. AC		**HAM vs. U**		**AC vs. U**	
Peak	q-value	Peak	q-value	Peak	q-value
11.7	7.00E-05	11.7	4.71E-06	-	-
11.9	2.38E-04	11.9	5.04E-05		
13.3	5.50E-04	13.3	1.16E-04		
14.6	1.49E-02	14.6	8.55E-04		
17.3	3.76E-02				

Table 3.1: Peaks identified as significant (after multiple testing correction) from the D_O data set, together with corresponding q-values.

HAM vs. AC		**HAM vs. U**		**AC vs. U**	
Peak	q-value	Peak	q-value	Peak	q-value
14.6	4.18E-02	14.6	2.39E-03	-	-
11.7	4.18E-02	11.7	4.45E-03		
13.3	4.18E-02	13.3	4.45E-03		
11.9	4.57E-02	17.5	1.05E-02		
		11.9	4.37E-02		
		13.7	4.38E-02		
		17.3	4.38E-02		
		59	4.38E-02		
		90.6	4.38E-02		
		8.58	4.4s1E-02		

Table 3.2: As in Table 3.1, using the data from the D_V data set.

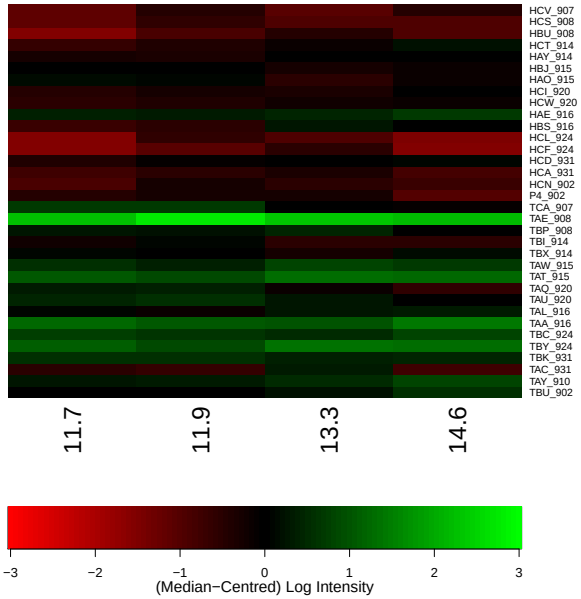


Figure 3.2: Heatmap representation of the pattern of intensities for the 11.7kDa, 11.9kDa, 13.3kDa and 14.6kDa peaks, using data from the original set. Rows correspond to different samples (with the top 17 being AC samples, and the bottom 17 being HAM), while columns correspond to different peaks. The colours of the blocks describe the log peak intensities (after median centring).

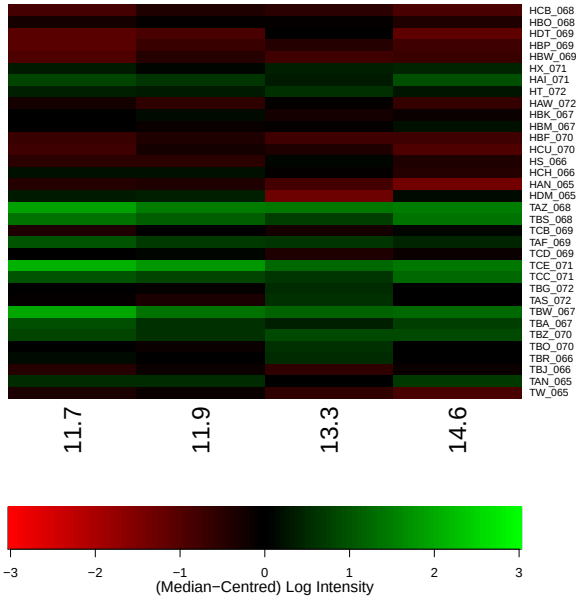


Figure 3.3: As in Figure 3.2, but this time showing the results from the verification data set. Again, the top 17 rows correspond to AC samples, and the bottom 17 to HAM.

3.4 Selection by stability and predictive performance

Having completed a preliminary statistical analysis, we now apply the algorithm of Chapter 2, Section 2.5. As previously mentioned, we believe assessments of selection stability to be particularly appropriate for SELDI-TOF MS data sets, given the concerns in the literature regarding reproducibility of results (Baggerly *et al.*, 2004). It is also widely recognised that strong correlations are characteristic of mass spectrometry data sets (Coombes *et al.*, 2003; Hand, 2008; Zuber and Strimmer, 2009), so it is important to account for their effects.

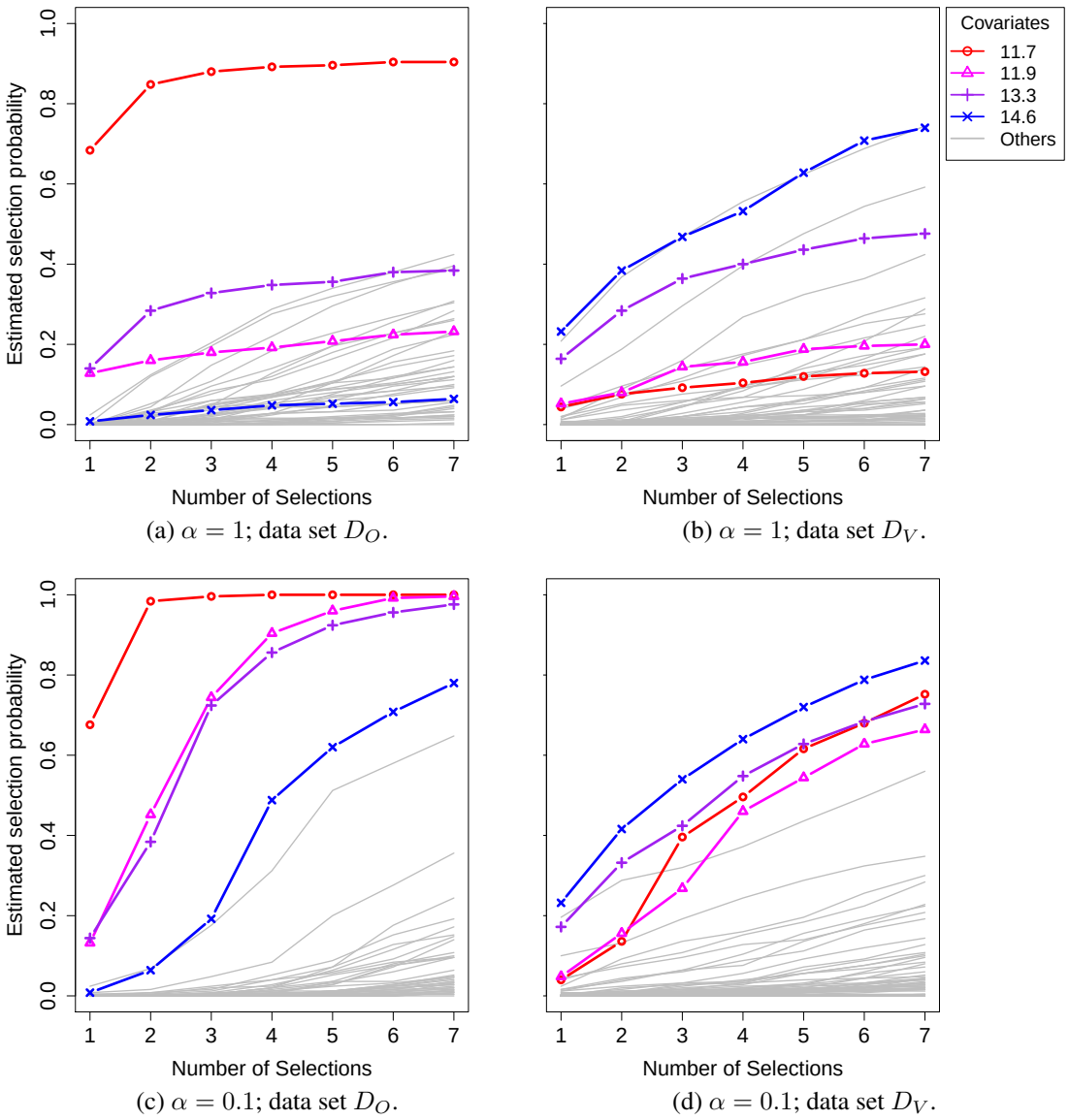
In Section 3.4.2, we apply our algorithm to the combined D_C data set. We focus on this data set rather than D_O or D_V, as we are principally interested in the differences between the HAM and AC classes, and so it is sensible to make use of all available HTLV-1 data. Given the randomised experimental procedure (Section 3.2), the normalisation for chip-specific effects (Section 3.2.1), and the good agreement between the results from the D_O and D_V data sets found in Section 3.3, we believe the combination of the HTLV-1 data sets to be justified.

Before embarking upon our main analysis, we make use of the D_O and D_V data sets once more. In Section 3.4.1, we demonstrate that the effects of correlation present difficulties for stability not only in the simulated example of Section 2.4.2, but also in real experimental examples.

3.4.1 Effects of correlation upon stability: a real example

For the D_O and D_V sets separately, we use a subsampling approach in order to estimate marginal selection probabilities for the protein peaks. We employ the same subsampling procedure as in Section 2.6, and again use logistic regression models with the elastic net likelihood penalty in order to make selections. We combine ideas of Sections 2.4.2 and 2.5 and plot stability paths that describe how the estimated marginal selection probabilities vary as a function of m (the number of selections), rather than λ (the regularisation parameter). We show in Figure 3.4 the resulting stability paths for D_O when the elastic net mixing parameter, α is set to 1 (Figure 3.4a), and to 0.1 (Figure 3.4c). Similar plots are shown for D_V in Figures 3.4b and 3.4d.

The figures show that when α is taken to be 1, there is little agreement between the stably selected covariates from the two data sets (for example, Figure 3.4a shows the 11.7kDa peak to be the most stably selected and the 14.6kDa peak to be barely ever selected; while in Figure 3.4b this situation is reversed). However, when the value for α is reduced to 0.1 (and hence correlated variables are permitted to be selected together), the agreement between the two plots is greatly increased, with the top 4 most stably selected peaks being in agreement (11.7kDa, 13.3kDa, 14.6kDa and 11.9kDa).



(a) $\alpha = 1$; data set D_O.

(b) $\alpha = 1$; data set D_V.

(c) $\alpha = 0.1$; data set D_O.

(d) $\alpha = 0.1$; data set D_V.

Figure 3.4: Stability paths for the D_O and D_V data sets, for values $\alpha = 1$ and $\alpha = 0.1$. Four covariates of interest are shown in colour.

The disagreement between Figures 3.4a and 3.4b, and the improved agreement as we reduce α, is due to strong correlations amongst the (putatively) relevant covariates. The correlation structure is represented visually in Figure 3.5, where we can see that the intensities of the 11.7, 13.3, 14.6 and 11.9kDa peaks are all strongly correlated with one another. This provides a real example of the effects that strong correlations can have upon selection stability.

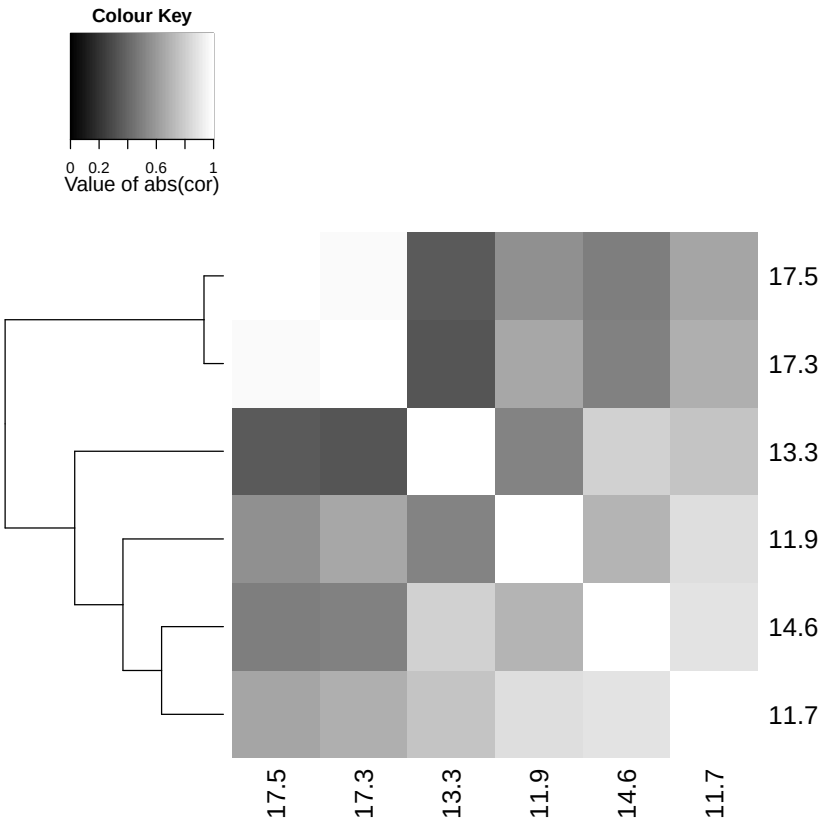


Figure 3.5: Heatmap representation of the correlation structure amongst 6 protein peaks in the D_C data set. The colour of each square indicates the magnitude of the (Pearson) correlation coefficient between each pair of protein peaks, as described by the colour key in the top left.

3.4.2 Applying Algorithm 1 to D_C

We apply our "joint score" selection method to the HTLV-1 combined data set, D_C, according to the approach presented in Chapter 2, Algorithm 1, and using the same subsampling procedure as in Section 2.6. We again employ logistic regression models with the elastic net penalty in order to select protein peaks, considering $\alpha = 0.1, 0.2, \ldots, 1$ (as in Section 2.6.1.1).

The final set returned by the algorithm comprises the protein peaks located at 11.7kDa, 13.3kDa and 14.6kDa (see Figure 3.6). We additionally show in Figure 3.7 the joint scores and estimated correct classification probabilities for 6 of the highest scoring sets. Each point corresponds to a selected set (as indicated by the legend). Moreover, each set may be represented up to ten times in the figure, as we display results from all of the values considered for α (if a set is represented fewer than ten times, this indicates that for some values of α, the set was never selected). We can see that sets comprising just the 11.7kDa or 13.3kDa peaks individually (respectively the red and magenta points) are amongst the highest scoring, and that all other high-scoring sets also include these two

covariates. The highest estimated probabilities of correct classification (around 0.78) are achieved for the set comprising both the 11.7kDa and 13.3kDa peaks (and no others). However, the highest joint scoring set additionally includes the 14.6kDa peak, indicating that selection of this covariate can help to improve stability. The intensities of all 3 of these peaks are correlated, and our results reflect this. Other high-scoring sets additionally include the 11.9kDa, 17.3kDa and 17.5kDa protein peaks (note that these covariates give rise to the local maximum in Figure 3.6). The 11.9kDa peak is strongly correlated with the 11.7kDa, 13.3kDa and 14.6kDa peaks, while the 17.3kDa and 17.5kDa peaks are particularly strongly correlated with each other (see Figure 3.5).

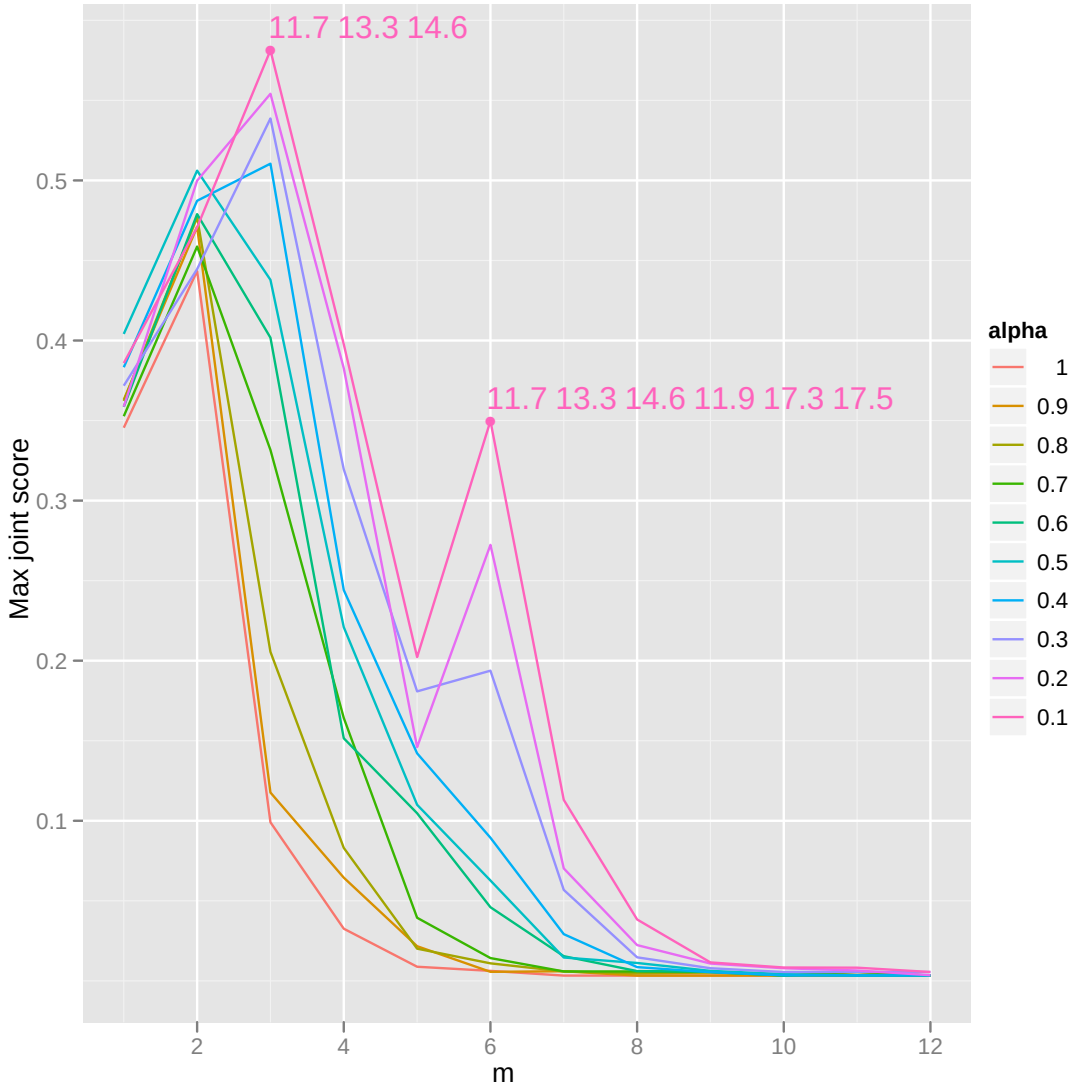


Figure 3.6: Maximal joint scores as a function of m for the D_C data set (cf. Figure 2.5). The set comprising the 11.7kDa, 13.3kDa and 14.6kDa peaks achieves the maximal joint score over all values of α, and hence is the final output of Algorithm 1.

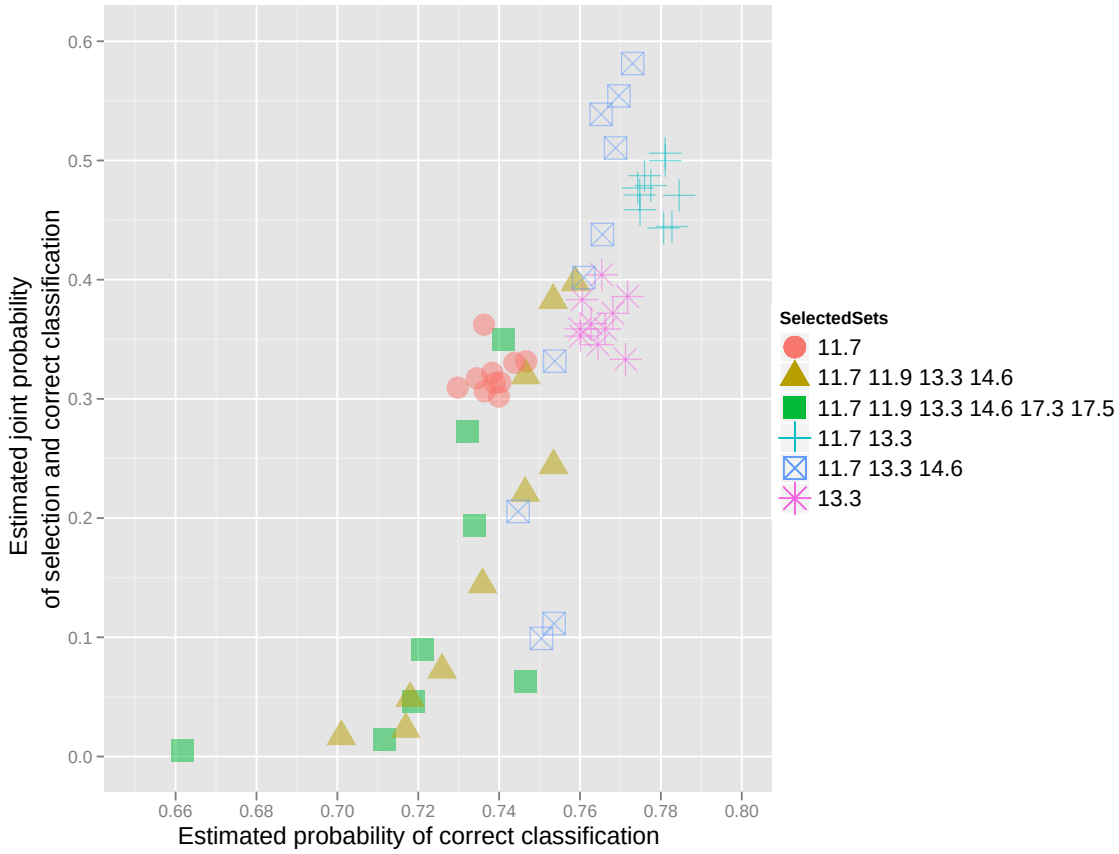
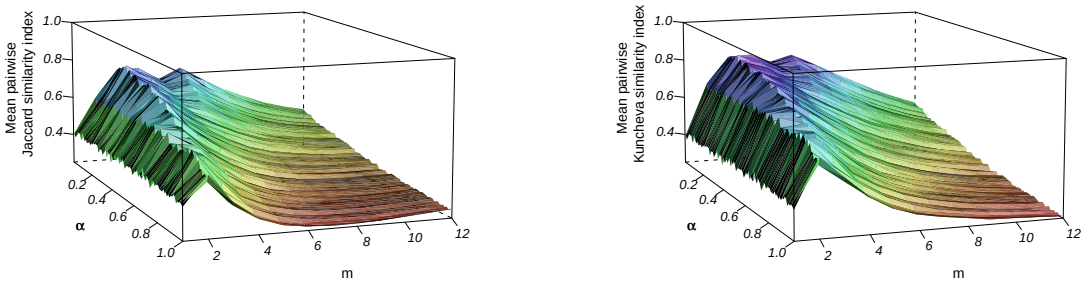


Figure 3.7: Joint score versus estimated probability of correct classification for 6 of the top scoring covariate sets, and for all considered values of α

3.4.2.1 Jaccard and Kuncheva similarity paths

For completeness, we also plot similarity paths (as in Figure 2.6 from Chapter 2), to show how the mean pairwise Jaccard and Kuncheva similarity coefficients vary as a function of m. We use the same subsampling scheme and predictive models as previously, but this time consider a finer grid of values for the elastic net mixing parameter, taking $\alpha = 0.01, 0.02, \ldots, 1$. We are hence able to plot similarity *surfaces*, showing how the similarity coefficients vary as a function of m and α, as shown in Figure 3.8. We first note that Figures 3.8a and 3.8b bear a strong resemblance to one another. The main difference between these two plots is that we can begin to see the Jaccard coefficient increasing for large values of m, while the Kuncheva coefficient corrects for this. There are local maxima in both plots at $m = 2$, $m = 3$ and $m = 6$, depending on the values of α. Although impossible to tell from just the similarity coefficients, Figure 3.6 allow us to interpret these maxima. The maximum at $m = 2$ corresponds to the stable selection of the 11.7kDa and 13.3kDa peaks; the maximum at $m = 3$ corresponds to the additional stable selection of the 14.6kDa peak; and the maximum at $m = 6$ corresponds to the additional stable selection of the 11.9kDa, 17.3kDa and 17.5kDa peaks.



(a) (b)

Figure 3.8: (a) Jaccard and (b) Kuncheva similarity surfaces for the D_C data set.

3.5 Experimental identification of protein peaks

The 11.7kDa and 13.3kDa protein peaks were identified using peptide mass fingerprinting techniques (see, for example Pappin *et al.*, 1993). The 11.7kDa peak was found to correspond to β_2-microglobulin, while the 13.3kDa peak corresponds to Calgranulin B. There is strong supporting evidence for the validity of these proteins as biomarkers. The β_2-microglobulin protein is a known biomarker of rheumatoid arthritis (Manicourt *et al.*, 1978) and other chronic inflammatory diseases (Xie and Yi, 2003), while serum Calgranulin B levels have been found to be elevated in a number of inflammatory disorders (Foell and Roth, 2004; Kelly *et al.*, 1989), including cystic fibrosis (Wilkinson *et al.*, 1988).

3.6 Discussion

We analysed SELDI-TOF MS data in order to identify putative biomarkers of HAM/TSP. We used a number of techniques to analyse these data, including the algorithm of Chapter 2, Section 2.5. Three protein peaks were returned by our algorithm, which were in good agreement with those selected using more traditional analyses (Section 3.3). Amongst these, there were two particular protein peaks (11.7kDa and 13.3kDa) that together provided the maximal cross-validation correct classification rate (around 78%) when discriminating between HAMs and ACs. These peaks were identified using peptide mass fingerprinting techniques, and were found to correspond to plausible protein biomarkers. Experimental identification of the remaining protein peak returned by our algorithm (14.6kDa) is ongoing. Although not returned amongst the final selections of our algorithm, efforts are also being made to identify at least one of the 17.3 and 17.5kDa peaks. Figure 3.6 provides weak evidence to suggest that these protein peaks might be important. Moreover, in our preliminary analyses (Section 3.3), we found them to be very close to the threshold FDR level, although usually falling just below it. In contrast to the 11.7kDa, 11.9kDa, 13.3kDa and 14.7kDa peaks, the intensities of the 17.3kDa and 17.5kDa peaks

are elevated amongst the ACs, and hence they might provide different insights into the pathogenesis of HAM/TSP.

By plotting stability paths for the D_O and D_V data sets (Figure 3.4), we demonstrated that the problems identified in Section 2.4.2 can occur in practice, as well as in simulated examples. Our algorithm, which allows us to consider a range of values for the mixing parameter of the elastic net likelihood penalty, allows these issues to be mitigated.

We found the plot shown in Figure 3.7 (where the joint score is plotted against the estimated probability of correct classification) to be particularly useful when deciding upon the order in which to identify the protein peaks. Having access to an assessment of predictive performance as well as an assessment of stability allowed us to target the first of the follow-up experiments toward the protein peaks that we believe will (individually and jointly) provide the greatest ability to classify.

Although they provide good summaries of selection stability, we again found the mean pairwise Jaccard and Kuncheva similarity indices to be of limited use when trying to determine exactly which covariates are stably selected. However, we were able to shed light on the causes of the local maxima in the Jaccard and Kuncheva similarity surfaces (Figure 3.8) by referring to the plot of maximal joint scores (Figure 3.6).

Overall, the quantities calculated by the algorithm of Chapter 2, Section 2.5, proved very useful, and provided deeper insights into the stability and predictive properties of the covariates than would be possible using either the stability selection procedure of Meinshausen and Bühlmann (2010) or standard similarity measures such as the Jaccard and Kuncheva indices. This is due both to our use of the joint score (to combine assessments of stability and predictive performance), and also the fact that we have a score associated with each set of selected covariates (rather than having a marginal score associated with single covariates, or an overall summary of stability).

There is a great deal of scope for further work following on from this chapter. In particular, blood plasma samples from multiple sclerosis patients have become available, and we are beginning to work with these. Identifying protein peaks that allow multiple sclerosis patients to be distinguished from individuals with HAM/TSP would potentially be of significant diagnostic value, as these two conditions can be difficult to tell apart (Bangham *et al.*, 1989; Rudge *et al.*, 1991). As previously mentioned, experimental work is also ongoing in order to identify a few more of the protein peaks identified during our analysis.

Chapter 4

A bootstrap for time course data

Overview

Abstract *Having previously considered assessments of stability for static data, we move on to a method for assessing the robustness of estimates obtained from time courses of measurements. We propose an approach in which we approximate the data generating process (DGP) as a multivariate Gaussian distribution. We consider two conceptually different Bayesian approaches for inferring the parameters of this distribution, and demonstrate that they both lead to the same approximate DGP.*

Outline *This chapter is concerned with a method for bootstrapping regression models. In Section 4.1, we briefly review an existing technique for addressing this task, before providing motivation for our multivariate Gaussian approach. We then describe in Section 4.2 a (finite-dimensional) conjugate Bayes approach in which we place a multivariate Gaussian prior over the values of the unknown noiseless values of the observations. After this, we discuss the machine learning approach of* Gaussian process regression *in Section 4.3, and thereby derive in Section 4.4 a (infinite-dimensional) Bayesian approach in which a Gaussian process prior is placed over the regression function f. Although these derivations are slightly different, both lead to the same approximate DGP (Section 4.5). Finally, in Section 4.6, we briefly discuss our procedure in the context of other bootstrap approaches.*

In Chapters 5 and 6 we consider two different applications of the approach described here.

4.1 Background

One of the limitations of the previous chapters is the static nature of the data considered. Although such data allow us to make useful statements regarding the general differences between two (or more) conditions, they provide no information regarding (for example) how the severity of symptoms of HAM/TSP change over time, or about the dynamics of the interactions between HTLV-1 and the immune system. In order to address such questions — and, crucially, to move beyond simple statistical models toward more mechanistic representations of biological systems — we require time-resolved measurements (Grigorov, 2006; Sato *et al.*, 2008). Such time course data are known to present challenges for bootstrapping approaches, as it is necessary to ensure that correlations between measurements at different time points are adequately captured (Efron and Tibshirani, 1993; Bühlmann, 2002). In this chapter, we address this problem in the context of bootstrapping using regression models (Efron and Tibshirani, 1993; Davison and Hinkley, 1999).

4.1.1 Bootstrapping using regression models

We consider regression models of the form,

$$y(t) = f(t) + \epsilon, \tag{4.1}$$

where $f(t)$ is a deterministic function of the covariates, and ϵ is a zero-centred random variable. The observations that we obtain through experimentation are viewed as realisations of the random variable $y(t)$. Throughout, we only consider regression problems in which $t \in \mathbb{R}$ is time, and we further restrict ourselves to the case where $f(t)$ and $y(t)$ are real-valued. Unless otherwise stated, we also assume that ϵ is normally distributed, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$.

We initially assume that our data comprise measurements taken at p distinct time points, $t_1, \ldots, t_p$, and that at each t_i we have a single observation, y_i. Current biological time course data are often of this form, although we later consider situations in which we have more than one observation at each time point (see Chapter 5). In traditional approaches to regression, we choose a parametric form for f, whose parameters we then estimate by fitting to the observed data. We denote the fitted function by $\hat{f}$. For example, in simple linear regression we would assume $f(t) = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 t$, and would obtain estimates $\hat{\boldsymbol{\beta}}_0$ and $\hat{\boldsymbol{\beta}}_1$ of the coefficients by (for example) ordinary least squares. The fitted function would then be $\hat{f}(t) = \hat{\boldsymbol{\beta}}_0 + \hat{\boldsymbol{\beta}}_1 t$.

Bootstrapping residuals Given such a regression model, the most common method for obtaining replicate data sets is to *bootstrap the residuals* (Efron and Tibshirani, 1993). This requires us to start by calculating the residuals of the fitted model,

$$\hat{\varepsilon}_i = y_i - \hat{f}(t_i), \tag{4.2}$$

for $i = 1, \ldots, p$, in order to obtain the set $\mathcal{E} = \{\hat{\varepsilon}_i\}_{i=1}^p$. We then use the standard methods of Chapter 1 in order to obtain a bootstrap sample of the residuals, $\mathcal{E}_B = \{\hat{\varepsilon}_i^*\}_{i=1}^p$. For example, in the nonparametric case, $\mathcal{E}_B$ would be formed by sampling from $\mathcal{E}$ with replacement. It is then straightforward to obtain bootstrap samples of our observations as,

$$y_i^{\text{rep}} = \hat{f}(t_i) + \hat{\varepsilon}_i^*, \tag{4.3}$$

for $i = 1, \ldots, p$. This procedure is illustrated in Figure 4.1. We may repeat the process many times in order to obtain a large number of replicate data sets.

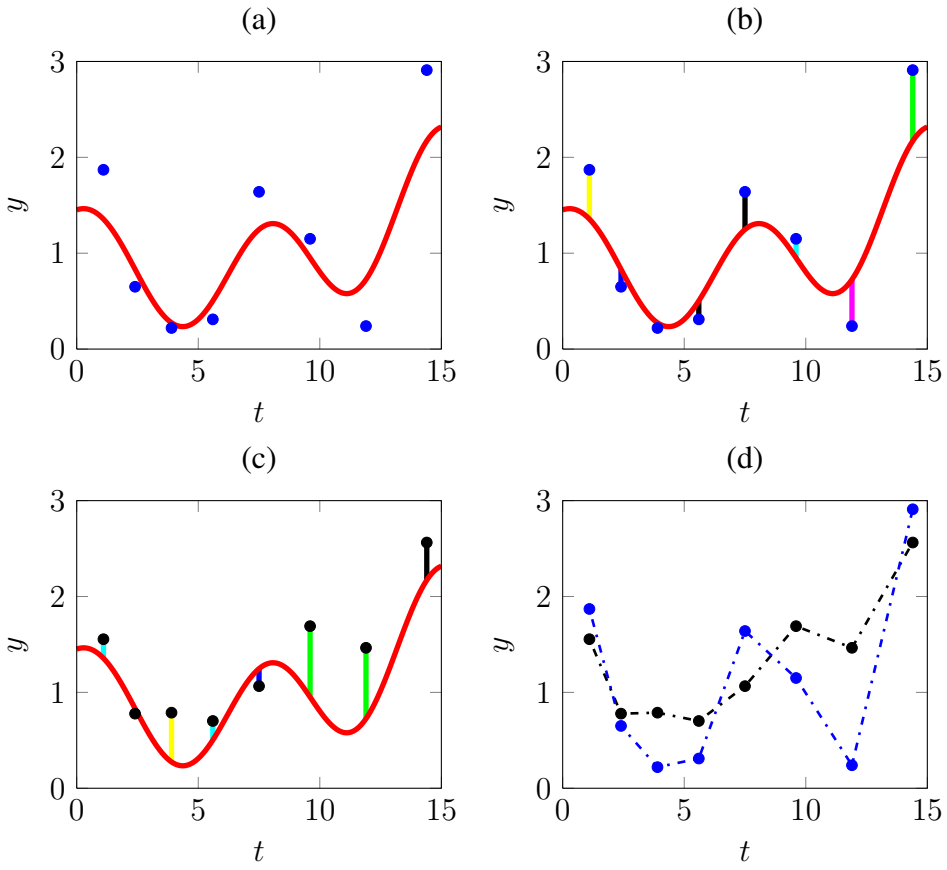


Figure 4.1: Illustration of a nonparametric bootstrap of residuals. (a) The blue points represent the original data set, while the red line denotes the fit provided by a particular model. (b) We calculate the residuals, here represented as vertical lines drawn between the observed data and the fitted model. (c) We perform a nonparametric bootstrap on the residuals by sampling with replacement from the set of vertical lines. By adding these to our fitted model, we obtain a bootstrap data set (the black points). (d) The original (blue) and bootstrap (black) data set, with connecting dashed lines added to aid visualisation.

The main difficulty with the above method is that it relies upon the fitted parametric model, $\hat{f}$. In systems biology problems, it is often difficult to establish an appropriate form for f, with mechanistic ("knowledge-driven") specifications typically being both uncertain and incomplete. One alternative is to fit an empirical ("data-driven") regression

model — using, for example, ANOVA models (as in Kerr and Churchill, 2001) or a more flexible approach such as an artificial neural network — and then to obtain replicate data sets as above, by bootstrapping residuals. However, regardless of the regression method we employ, we are again overly reliant on a single fitted model, $\hat{f}$. In practice, we will rarely have complete belief that $\hat{f}$ is identical to the true, underlying function f. We hence propose an approach in which we explicitly model the uncertainty in the values of $f(t_i), i = 1, \ldots, p$.

4.1.2 Motivation and chapter outline

In this chapter, we propose a multivariate Gaussian bootstrap procedure for time course data. We view the values of $f(t_i)$, $i = 1, \ldots, p$, as unknown parameters, and hence rephrase our regression problem in terms of the discrete model,

$$\mathbf{y}^\top = \boldsymbol{\theta}^\top + \boldsymbol{\varepsilon}^\top, \tag{4.4}$$

where $\mathbf{y} = [y_1, \ldots, y_p]$ is the vector of observed values for y; $\boldsymbol{\theta} = [f(t_1), \ldots, f(t_p)]$ is the vector of unknown values for $f(t_i)$; and $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_p]$ is the vector of residuals. By assumption, the discrepancy between $\mathbf{y}$ and θ arises as a result of univariate normal experimental measurement noise, so that $\varepsilon_1, \ldots, \varepsilon_p$ may be viewed as independent samples from $\mathcal{N}(0, \sigma_\epsilon^2)$, and hence $\boldsymbol{\varepsilon}^\top$ is a sample from the multivariate normal $\mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 I)$.

In Section 4.2, we adopt a Bayesian approach toward the inference of $\boldsymbol{\theta}^\top$, specifying a multivariate normal prior, $\boldsymbol{\theta}^\top \sim \mathcal{N}(\mathbf{m}_o, \Sigma_o)$, and updating this in light of the observed data in order to obtain the posterior distribution. An informal motivation for why and how a multivariate Gaussian may be an appropriate prior for $\boldsymbol{\theta}^\top = [f(t_1), \ldots, f(t_p)]^\top$ is provided by Figure 4.2. At the heart of our approach is the use of a parametric *covariance function*, k, which specifies the elements of Σ_0 by modelling how the covariance between $f(t_i)$ and $f(t_j)$ varies as a function of t_i and t_j.

The use of a function k to capture the covariance structure of the data is identical to the approach taken in *Gaussian process regression* (GPR). GPR is a Bayesian nonparametric method that has grown in popularity in recent years, and has been applied in several systems biology contexts (Liu *et al.*, 2010; Stegle *et al.*, 2010; Lawrence *et al.*, 2007; Yuan, 2006; Gao *et al.*, 2008). In Section 4.3, we describe this method in more detail and then consider in Section 4.4 how GPR may be used to generate replicate data sets.

In the chapters following this one, we demonstrate our approach using two examples from the systems biology literature. In Chapter 5 we consider the inference of networks from the *Arabidopsis thaliana* data set of Smith *et al.* (2004), considering both relevance (Butte *et al.*, 2000) and partial correlation networks (e.g. Opgen-Rhein and Strimmer, 2007a). We then look in Chapter 6 at the problem of estimating the parameters of an ordinary differential equation model proposed by Swameye *et al.* (2003) for the STAT5 signalling pathway.

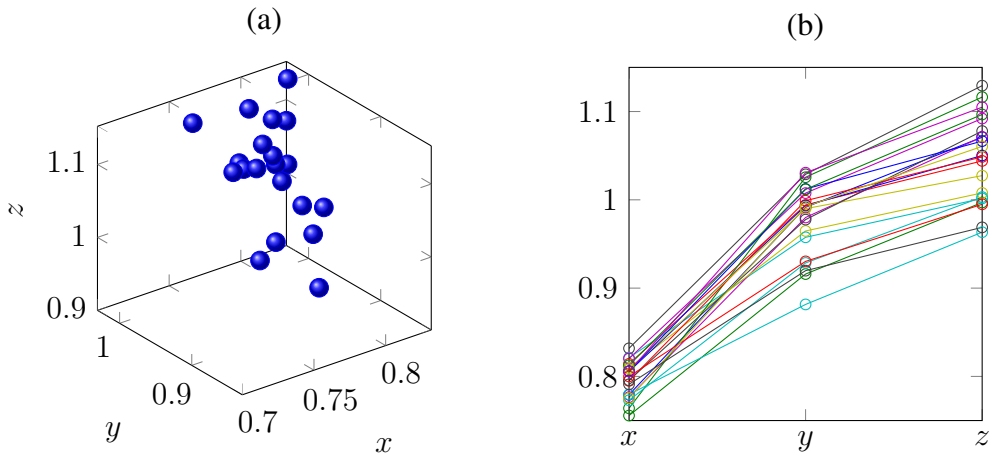


Figure 4.2: In (a) we show 20 samples drawn from a trivariate Gaussian distribution, depicted in a conventional way using a 3-d scatterplot. These data points may also be represented longitudinally as in (b). Here, each sample, $[x, y, z]^\top$, is represented by three points — one for each element of the vector — connected by a straight line. We are concerned with the reverse problem: given a single longitudinal time course comprising measurements at p time points, we seek to model this as a sample from a p-variate Gaussian distribution.

4.2 Multivariate Gaussian bootstrap

Recall our assumptions that we have a vector of measurements, $\mathbf{y} = [y_1, \ldots, y_p] \in \mathbb{R}^{1\times p}$, taken at times $t_1, \ldots, t_p$, and that each y_i may be represented as,

$$y_i = f(t_i) + \varepsilon_i. \tag{4.5}$$

Here, f is some continuous deterministic (but unknown) function of time, and we assume that the residuals, ε_i, are independently and identically distributed according to $\mathcal{N}(0, \sigma_\epsilon^2)$. This model allows us to express the assumption that there is some "true", underlying output that is deterministically linked to the value of t (as represented by $f(t)$), but that the output we actually observe has been corrupted by stochastic measurement noise. We may rewrite Equation (4.5) as the vector equation,

$$\mathbf{y}^\top = \boldsymbol{\theta}^\top + \boldsymbol{\varepsilon}^\top, \tag{4.6}$$

where $\boldsymbol{\theta} = [f(t_1), \ldots, f(t_p)]$ and $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_p]$. Traditional regression approaches would seek to estimate $\boldsymbol{\theta}$ by fitting a parametric model, $\hat{f}$, to the data, and then taking $\boldsymbol{\theta} = [\hat{f}(t_1), \ldots, \hat{f}(t_p)]$.

4.2.1 A Bayesian approach

An alternative strategy is to regard $\boldsymbol{\theta}$ simply as an unknown parameter of the model presented in Equation (4.6). We may then adopt a Bayesian approach to the inference of the

vector $\boldsymbol{\theta}$. In order to do this, we must first define a *prior*, $p(\boldsymbol{\theta})$, and a *likelihood function*, $p(\mathbf{y}|\boldsymbol{\theta})$, where $\mathbf{y} = [y_1, \ldots, y_p]^\top$ is the vector of observations. The prior describes our belief about the values of $f(t_1,), \ldots, f(t_p)$ *before* observing the data, while the likelihood function scores different possibilities for $\boldsymbol{\theta}$ by defining the likelihood of $\mathbf{y}$ for any given choice of $\boldsymbol{\theta}$. Bayes rule then provides a principled means to update our prior beliefs in light of the observed data,

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})}{p(\mathbf{y})}. \tag{4.7}$$

Here, $p(\boldsymbol{\theta}|\mathbf{y})$ is the *posterior*, which describes our belief about the values of $f(t_1,), \ldots, f(t_p)$ *after* observing the data, and $p(\mathbf{y})$ is the *marginal likelihood* (sometimes also called the *prior predictive distribution*). The marginal likelihood is a constant term defined as,

$$p(\mathbf{y}) = \int p(\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{4.8}$$

where the integral is taken over all possibilities for $\boldsymbol{\theta}$. The presence of this term in the denominator of Equation (4.7) ensures that $\int p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} = 1$.

4.2.2 Defining the prior

Motivated in part by the observations of Figure 4.2, we take our prior to be a p-dimensional multivariate Gaussian,

$$\boldsymbol{\theta}^\top \sim \mathcal{N}(\mathbf{m}_0, K_0). \tag{4.9}$$

For simplicity, we shall always take $\mathbf{m}_0$ to be the zero vector. In practice, this choice is not overly restrictive (in particular, we shall see that it does not constrain the mean of the posterior to be zero), but it does simplify some of the calculation. However, for the sake of generality and to allow for the possibility of stronger prior information, we shall leave $\mathbf{m}_0$ as an unspecified vector of length p in our initial exposition.

The choice of K_0 is perhaps more important. In practice, specifying K_0 according to our prior beliefs is likely to be challenging, since we may know very little about the nature of the unknown function f. However, if we assume that f is continuous, then we do at least have the prior belief that $f(t_i)$ and $f(t_j)$ should be more strongly correlated if t_i and t_j are close together, and should be less strongly correlated if they are far apart. This motivates the use of a *covariance function*, k, which defines the covariance between $f(t_i)$ and $f(t_j)$ as a function of t_i and t_j, so that,

$$(K_0)_{ij} = k(t_i, t_j). \tag{4.10}$$

As we shall discuss, such functions are at the heart of the machine learning method of *Gaussian process regression* (GPR). One choice for k that is commonly encountered in the GPR literature is the *squared exponential* covariance function,

$$k_{SE}(t_i, t_j) = \sigma_g^2 \exp\left(-\frac{(t_i - t_j)^2}{2\ell}\right), \tag{4.11}$$

where σ_g^2 and ℓ are parameters of the covariance function, referred to as *hyperparameters*. Other choices of covariance function are possible, all of which introduce hyperparameters. Note, however, that our use of a covariance function has simplified our problem: we no longer have to specify the whole covariance matrix, K_0, but instead have only to specify σ_g^2 and ℓ (in the squared exponential case). We discuss the choice of hyperparameters in Section 4.2.4.

4.2.3 Defining the likelihood

Recall our assumption that the residuals, ε_i, are independently and identically distributed according to $\mathcal{N}(0, \sigma_\epsilon^2)$. It then follows from Equation (4.5) that,

$$\mathbf{y}|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma_\epsilon^2 I), \tag{4.12}$$

where I is the $p \times p$ identity matrix. The appropriate form for the likelihood is then given by the usual multivariate normal probability density function, with mean $\boldsymbol{\theta}$ and covariance $\sigma_\epsilon^2 I$,

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{1}{(2\pi)^{p/2}|\sigma_\epsilon^2 I|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \boldsymbol{\theta})^\top (\sigma_\epsilon^2 I)^{-1}(\mathbf{y} - \boldsymbol{\theta})\right). \tag{4.13}$$

4.2.4 Selecting the hyperparameters

From Equations (4.9) and (4.12) it follows that,

$$\mathbf{y} \sim \mathcal{N}(\mathbf{m}_0, K_0 + \sigma_\epsilon^2 I), \tag{4.14}$$

and hence the marginal likelihood, $p(\mathbf{y})$, is simply,

$$p(\mathbf{y}) = \frac{1}{(2\pi)^{p/2}|K_0 + \sigma_\epsilon^2 I|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{m}_0)^\top (K_0 + \sigma_\epsilon^2 I)^{-1}(\mathbf{y} - \mathbf{m}_0)\right). \tag{4.15}$$

Note that this may also be calculated by explicit evaluation of Equation (4.8). We adopt the strategy of estimating the hyperparameters and σ_ϵ^2 by finding the values that maximise this marginal likelihood, or — equivalently — minimise the negative log marginal likelihood (Rasmussen and Williams, 2005). The estimates are then given by,

$$\widehat{\boldsymbol{\beta}}, \widehat{\sigma_\epsilon^2} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ (\mathbf{y} - \mathbf{m}_0)^\top (K_0(\boldsymbol{\beta}) + \sigma_\epsilon^2 I)^{-1}(\mathbf{y} - \mathbf{m}_0) + \log|K_0(\boldsymbol{\beta}) + \sigma_\epsilon^2 I| \right\}, \tag{4.16}$$

where $\boldsymbol{\beta}$ is just a shorthand for the vector of all hyperparameters (e.g. $\boldsymbol{\beta} = [\sigma_g^2, \ell]^\top$ in the case of the squared exponential function of Equation 4.11), and we make clear

the dependence of K_0 upon $\boldsymbol{\beta}$ by writing $K_0(\boldsymbol{\beta})$. Solving this minimisation problem requires numerical optimisation techniques; we use Rasmussen's implementation of a Polak-Ribière conjugate gradient scheme (Rasmussen, 2006).

The above procedure represents an *empirical Bayes* approach, in which the hyperparameters defining the prior are estimated from the data (Bernardo and Smith, 1994). An alternative is to define priors for the hyperparameters, and then to use a Markov chain Monte Carlo method in order to perform Bayesian inference (e.g. Barber and Williams, 1997). Although appealing, such approaches are more computationally costly, so we choose not to pursue them here.

4.2.5 Evaluating the posterior

Once we have estimated σ_ϵ^2, it is clear from Equation 4.12 that the problem of inferring $\boldsymbol{\theta}$ is equivalent to the well-known problem of inferring the mean of a multivariate normal distribution when the covariance matrix is known. Furthermore, since our prior for $\boldsymbol{\theta}$ is multivariate normal, and hence is a *conjugate* prior for the multivariate normal likelihood, it follows that we can calculate the posterior analytically (see, for example, Gelman, 2004), to give,

$$\boldsymbol{\theta}|\mathbf{y} \sim \mathcal{N}(\mathbf{m}_n, K_n), \tag{4.17}$$

where,

$$\mathbf{m}_n = \left(K_0^{-1} + \frac{1}{\sigma_\epsilon^2} I\right)^{-1} \left(K_0^{-1}\mathbf{m}_0 + \frac{1}{\sigma_\epsilon^2}\mathbf{y}\right) \tag{4.18}$$

and,

$$K_n = \left(K_0^{-1} + \frac{1}{\sigma_\epsilon^2} I\right)^{-1}. \tag{4.19}$$

4.2.6 Defining the approximate DGP

From Equations (4.12) and (4.17), we deduce that an appropriate model for our approximate DGP is,

$$\mathbf{y}^{\text{rep}} \sim \mathcal{N}(\mathbf{m}_n, K_n + \sigma_\epsilon^2 I). \tag{4.20}$$

It is clear that this may be viewed as a two-stage procedure in which we first use Equation (4.17) in order to sample values for the unknown function outputs $\boldsymbol{\theta} = [f(t_1), \ldots, f(t_p)]$, and then use Equation (4.12) to add simulated measurement noise.

As we shall now discuss, with small amendments to our assumptions, the above method may be adapted to model the uncertainty in the *function* f — not merely the vector $\boldsymbol{\theta}$ — and our approach may be derived from the perspective of the machine learning approach of *Gaussian process regression*.

4.3 Gaussian process regression

Gaussian process regression (GPR) is a method for nonlinear regression. It may be viewed as a Bayesian approach in which we first place a *Gaussian process prior* over the regression function f, and then update this in light of observed data.

More formally, a Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution (Rasmussen and Williams, 2005). This definition simply means that a (potentially infinite) collection $v_1, v_2, \ldots$ of random variables defines a Gaussian process if and only if any finite subcollection of the variables is jointly distributed according to a Gaussian distribution. So, for example, v_1 by itself must be Gaussian distributed; the joint distribution of v_1 and v_2 must be Gaussian; ... and so on.

In this section, we explain how a GP prior may be specified and then updated in light of observed data in order to obtain a GP posterior. We note that the derivation presented in this section is rather similar to the one given in Section 4.2. However, with apologies for repetition, we nevertheless proceed, as we feel that it is instructive to consider the finite- and infinite-dimensional cases separately.

4.3.1 Defining the prior

In Gaussian process regression, a GP prior is assumed for the outputs of the unknown function f. This means that we assume $f(t_1), f(t_2), \ldots, f(t_r)$ to have a joint Gaussian distribution for any $t_1, t_2, \ldots, t_r$ and any finite r. In order to specify the Gaussian process prior, we require a mean function, m, and a covariance function, k. These tell us how to define the mean vectors and covariance matrices of the Gaussian distributions associated with each finite subcollection of the variables. We write $f \sim \mathcal{GP}(m, k)$ to indicate that we have assumed a Gaussian process prior with mean function m and covariance function k for the function f. This is shorthand for the following:

> We write $f \sim \mathcal{GP}(m, k)$ if and only if, for any finite collection $t_1, t_2, \ldots, t_r$ of times, we have $[f(t_1), \ldots, f(t_r)]^\top \sim \mathcal{N}(\mathbf{m}, K)$, where $\mathbf{m}_i = m(t_i)$ and $K_{ij} = k(t_i, t_j)$.

There are many possibilities for m and k. In practice, we shall always take m to be the zero function, so that $m(t_i) = 0$ for all t_i. As before, this is not excessively restrictive, but for the sake of generality we assume m to be a general mean function throughout our exposition. The covariance function is rather more important, as it allows us to express beliefs about the correlations between $f(t_i)$ and $f(t_{j\neq i})$. As previously mentioned, one popular choice for k is the squared exponential covariance function of Equation (4.11). As before, having chosen a particular parametric form for our covariance function, we must estimate its hyperparameters.

4.3.2 Estimating the hyperparameters

As previously stated, we assume a constant-variance, univariate normal noise model, so that, for all t, we have,

$$y(t) = f(t) + \epsilon, \tag{4.21}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. We assume a Gaussian process prior over $f(t)$, from which it follows that, for any finite collection ${t_1}^*, \ldots, {t_s}^*$ of times, we have,

$$\left[y\left({t_1}^*\right), \ldots, y\left({t_s}^*\right)\right]^\top \sim \mathcal{N}(\mathbf{m}_*, K_{**} + \sigma_\epsilon^2 I), \tag{4.22}$$

where $\mathbf{m}_* = [m(t_1^*), \ldots, m(t_s^*)]^\top$ and $(K_{**})_{ij} = k(t_i^*, t_j^*)$. In other words, with our assumed noise model, the Gaussian process prior over $f(t)$ induces a Gaussian process prior over $y(t)$,

$$y(t) \sim \mathcal{GP}(m_y, k_y), \tag{4.23}$$

where $m_y(t_i) = m(t_i)$ and $k_y(t_i, t_j) = k(t_i, t_j) + \sigma_\epsilon^2 \delta(t_i, t_j)$. Here, $\delta(t_i, t_j)$ is the Kronecker delta function which is equal to 1 whenever $t_i = t_j$ and 0 otherwise.

In particular,

$$\mathbf{y} \sim \mathcal{N}(\mathbf{m}_0, K_0 + \sigma_\epsilon^2 I), \tag{4.24}$$

where,

$$\mathbf{m}_o = [m(t_1), \ldots, m(t_r)]^\top, \text{ and} \tag{4.25}$$

$$(K_o)_{ij} = k(t_i, t_j). \tag{4.26}$$

Note that Equation (4.24) is identical to Equation (4.14), and hence we may adopt exactly the same approach to estimating our covariance function's hyperparameters as previously (see Equation 4.16).

4.3.3 Evaluating the posterior

Recall that we have observations made at $t_1, \ldots, t_r$. From our assumption of a Gaussian process prior for $f(t)$, it follows that, given any finite collection ${t_1}^*, \ldots, {t_s}^*$ of times, we have,

$$\left[f(t_1), \ldots, f(t_r), f({t_1}^*), \ldots, f({t_s}^*)\right]^\top \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{m}_o \\ \mathbf{m}_*\end{bmatrix}, \begin{pmatrix} K_o & K_{o*} \\ K_{*o} & K_{**}\end{pmatrix}\right), \tag{4.27}$$

where,

$$\mathbf{m}_* = [m({t_1}^*), \ldots, m({t_s}^*)]^\top, \tag{4.28}$$

$$(K_{o*})_{ij} = k(t_i, {t_j}^*), \tag{4.29}$$

$$(K_{*o})_{ij} = k({t_i}^*, t_j), \tag{4.30}$$

$$(K_{**})_{ij} = k({t_i}^*, {t_j}^*). \tag{4.31}$$

Hence, from Equation (4.22),

$$[y(t_1), \ldots, y(t_r), f(t_1{}^*), \ldots, f(t_s{}^*)]^\top \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{m}_o \\ \mathbf{m}_*\end{bmatrix}, \begin{pmatrix} K_o + \sigma_\epsilon^2 I & K_{o*} \\ K_{*o} & K_{**}\end{pmatrix}\right). \quad (4.32)$$

From Equation (4.32), and using standard properties of Gaussian distributions (von Mises, 1964), it follows that the function values $f(t_1{}^*), \ldots, f(t_s{}^*)$ conditioned on $y(t_1), \ldots, y(t_r)$ are also jointly distributed according to a multivariate normal. Specifically,

$$[f(t_1{}^*), \ldots, f(t_s{}^*)]^\top \Big| \, \mathbf{y}(t) \quad \sim \mathcal{N}(\mathbf{m}_{cond}, K_{cond}), \quad (4.33)$$

where $\mathbf{y}(t) = [y(t_1), \ldots, y(t_p)]^\top$, and,

$$\mathbf{m}_{cond} = \mathbf{m}_* + K_{*o}\left(K_o + \sigma_\epsilon^2 I\right)^{-1}(\mathbf{y}(t) - \mathbf{m}_o), \quad (4.34)$$

$$K_{cond} = K_{**} - K_{*o}\left(K_o + \sigma_\epsilon^2 I\right)^{-1} K_{o*}. \quad (4.35)$$

If we plug our observed output values into Equation (4.34) — i.e. we set $\mathbf{y}(t) = \mathbf{y}^\top$ — then Equation (4.33) defines the posterior distribution of $[f(t_1{}^*), \ldots, f(t_s{}^*)]^\top$, given our observations. Since $t_1{}^*, \ldots, t_s{}^*$ may be *any* finite collection of times, Equations (4.33) – (4.34) define a Gaussian process: the *Gaussian process posterior* for $f(t)$ given observations $\mathbf{y}$.

4.3.4 Visualisation

Equation (4.33) describes the joint posterior distribution of function outputs evaluated at any finite collection of times. In particular, we can take a single time, say t_*, and use Equation (4.33) to derive the posterior distribution for $f(t_*)$. This will clearly be a univariate normal whose mean, μ_*, and variance, σ_*^2, are given by Equations (4.34) and (4.35) respectively.

For any t_*, we can hence define the predictive mean value for $f(t_*)$ (namely, μ_*), and also standard deviation error bars (derived from σ_*^2). One way of visualising the Gaussian process posterior is then to consider a fine grid of values for t_*, and to plot the resulting predictive means, μ_*, together with $\pm$ 2 standard deviation error bars. An example of this is shown in Figure 4.3. Note that, strictly, the apparently continuous regression lines actually comprise a fine grid of discrete points (and similarly for the shaded ± 2 standard deviation region).

4.4 Gaussian process regression bootstrap

We now consider how the Gaussian process regression framework may be used to generate replicate data sets. Recall that Equation (4.33) holds for *any* finite collection of time

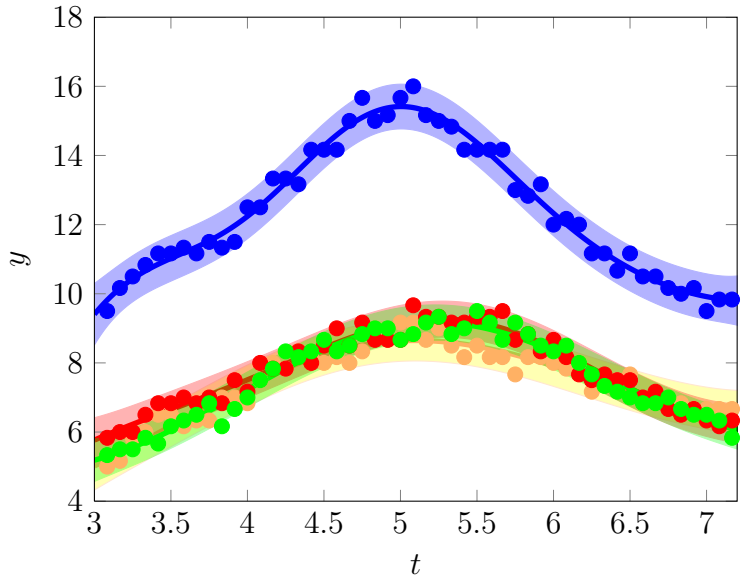


Figure 4.3: Four fitted Gaussian process regression models, as used in Taylor *et al.* (2010). The plot shows four different data sets (represented by the blue, red, green and yellow points), with independent Gaussian process regression models fitted to each set. The solid regression lines show how the pointwise predictive means vary as a function of time, while the shaded regions describe ± 2 standard deviation confidence intervals.

points, so in particular it holds for the time points $t_1, \ldots, t_p$ at which the observations were made. Of course, in this case, we will have $\mathbf{m}_* = \mathbf{m}_o$ and $K_{**} = K_{*o} = K_{o*} = K_o$, so that,

$$[f(t_1), \ldots, f(t_p)]^\top \Big| \mathbf{y} \quad \sim \mathcal{N}(\mathbf{m}_n, K_n), \tag{4.36}$$

where,

$$\mathbf{m}_n = \mathbf{m}_o + K_o \left(K_o + \sigma_\epsilon^2 I\right)^{-1} (\mathbf{y} - \mathbf{m}_o), \tag{4.37}$$

$$K_n = K_o - K_o \left(K_o + \sigma_\epsilon^2 I\right)^{-1} K_o. \tag{4.38}$$

We may hence generate replicate data sets by sampling values for $[f(t_1), \ldots, f(t_p)]^\top$ from $\mathcal{N}(\mathbf{m}_n, K_n)$, and then adding simulated measurement noise generated from $\mathcal{N}(0, \sigma_\epsilon^2)$, as illustrated in Figure 4.4. Of course, we may combine these two steps and sample replicate data sets directly according to,

$$\mathbf{y}^{\text{rep}} \sim \mathcal{N}(\mathbf{m}_n, K_n + \sigma_\epsilon^2 I). \tag{4.39}$$

In principle, we could generate replicate observations at any finite collection of time points. In practice, however, we sample only at the time points where the observations were made. This is partly for the sake of consistency with existing bootstrap procedures (such as bootstrapping residuals), but also because it is a more conservative approach (in the sense that it generates replicate data sets most similar to the original one).

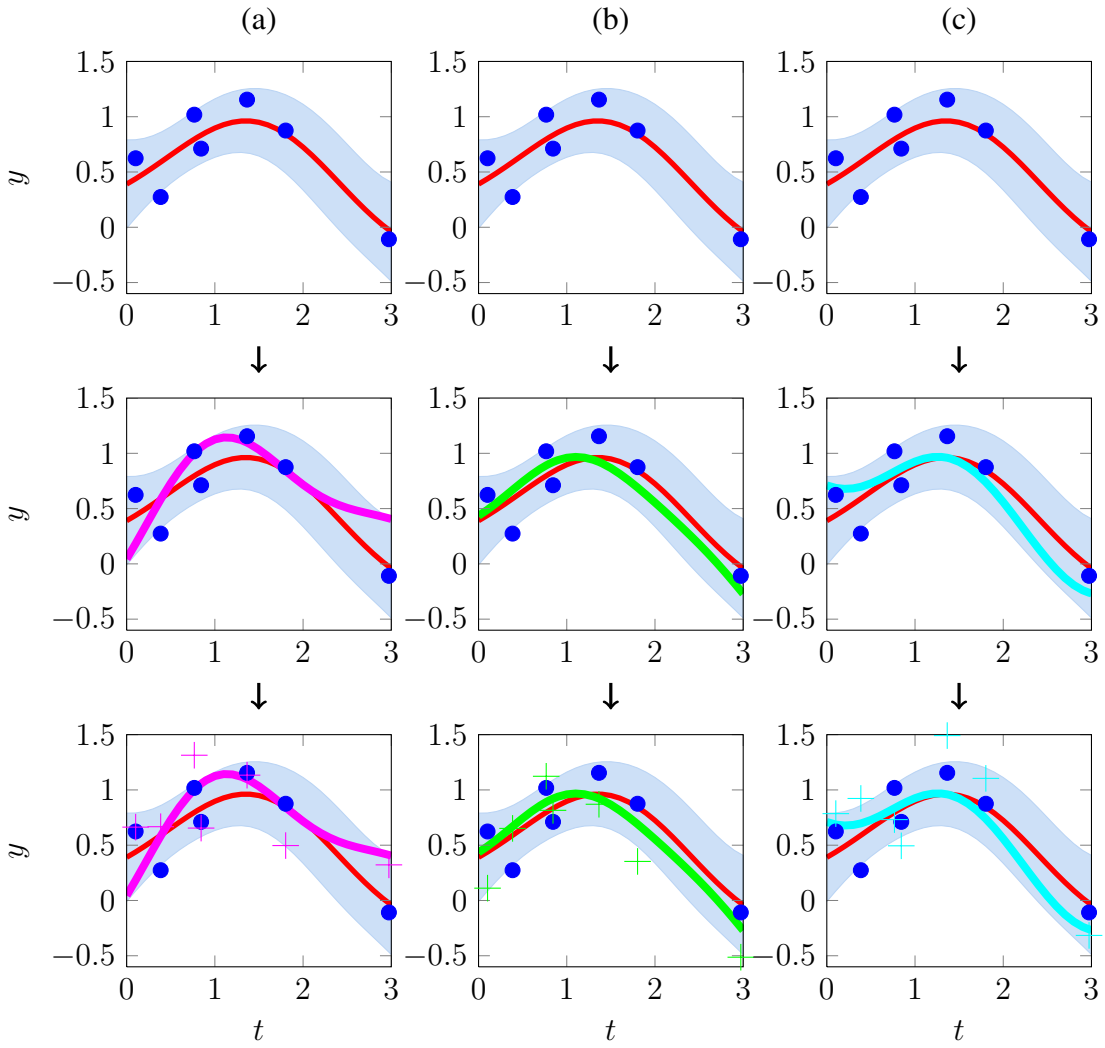


Figure 4.4: (a), (b) and (c) each demonstrate the procedure for drawing bootstrap samples from the Gaussian process regression model. In each case, the first row illustrates the Gaussian process regression model that has been fitted to the original data set; the second row shows a bootstrap *function* drawn from the GP posterior; and the third row shows the final replicate data set, obtained by adding noise to the sampled function.

4.5 Multivariate Gaussian versus GPR bootstrap

We have presented two very similar methods for approximating DGPs that generate time course data, derived from slightly different standpoints. In both cases, we end up obtaining replicate data sets by sampling from multivariate Gaussians (see Equations 4.20 and 4.39). It is clear that these two approaches must lead to equivalent models for the approximate DGP. Although the matrix identities which allow us to deduce the equivalence of Equations (4.38) and (4.19), and of Equations (4.37) and (4.18), are perhaps less obvious,

they follow immediately from the Woodbury matrix identity (see, for example Golub and Van Loan, 1996).

4.6 Discussion

We have proposed a multivariate Gaussian bootstrap for time course data. We derived the approximate DGP from two standpoints, considering both conjugate Bayesian and Gaussian process regression (GPR) perspectives.

As illustrated in Figure 4.4, our method may be considered as a two-stage approach, in which we first sample a function from the Gaussian process posterior, and then add noise simulated from $\mathcal{N}(0, \sigma_\epsilon^2)$. If, instead of sampling, we had a single, fixed estimate for the function, we would view our approach as a parametric bootstrap of the residuals, with the parametric model being specified by $\mathcal{N}(0, \sigma_\epsilon^2)$. It follows that our procedure may be considered as a natural extension of residual bootstrapping, in which we additionally model the uncertainty in the underlying regression function, f.

If we overlook the derivation and focus only on the final approximation to the DGP (Equation 4.20), we may view our approach as a straightforward multivariate Gaussian parametric bootstrap (albeit one in which a Bayesian approach is used to infer the parameters, instead of using maximum likelihood estimation). The Bayesian nature of our approach is crucial, however, as it allows us to specify prior belief regarding the covariance structure of the data via the covariance function. This enables us to overcome the difficulties associated with having so few observations. For example, we would typically have insufficiently many data points to estimate the entries of the covariance matrix in Equation (4.20) using a simple maximum likelihood method.

By analogy with the method of Rubin (1981b), it would seem appropriate to refer to our method as a *Bayesian parametric bootstrap*. Similar to Rubin, we have a model for the DGP (in our case a multivariate Gaussian, in Rubin's a Dirichlet distribution) whose parameters we infer using a conjugate Bayesian approach. An important contrast to Rubin's nonparametric procedure is that, as just discussed, the conjugate prior that we employ is necessarily informative.

In the next chapters, we apply our approach in the context of two inference problems commonly encountered in systems biology; namely, gene network inference (Chapter 5), and the estimation of parameters of ordinary differential equation models (Chapter 6).

Chapter 5

Application I: Gene network inference

Abstract *We apply the approach of Chapter 4 in the context of inferring networks from gene expression time course data. We focus upon the estimation of partial correlation networks from an Arabidopsis thaliana data set. We demonstrate that the inferred networks are relatively unstable, but that there are edges within the network that have a high level of bootstrap support.*

Outline *We start in Section 5.1 with a very brief introduction to networks, a description of the gene expression data set with which we are concerned, and a summary of the "GeneNet" algorithm of Schäfer* et al. *(2006). We then review the differences between* cross-sectional *and* longitudinal *time courses in Section 5.2, and consider how the nonparametric bootstrap and the method of Chapter 4 should be applied to these two types of data. We present results in Section 5.3, which we discuss in Section 5.4.*

5.1 Background

The use of networks to describe and model biological systems is now widespread (see, for example Barabasi and Oltvai, 2004; de Silva and Stumpf, 2005, and references therein). Applications include protein interaction (Kelly and Stumpf, 2008), metabolic (Ma and Zeng, 2003) and gene regulatory (de Jong, 2002; Schlitt and Brazma, 2007) networks. The underlying goal of network models is to describe dependencies (represented by edges) between covariates (represented by nodes). These might be physical dependencies (for example, we might construct a network of different molecular species, with edges drawn between those that have been experimentally determined to bind to one another), or statistical dependencies (for example, we might draw edges between genes whose expression levels are significantly correlated).

Given the complexity of biological systems, and the difficulties associated with making *in vivo* observations, elucidating statistical dependency networks is often a more realistic proposition than establishing all of the underlying (causal) physical interactions experimentally. Inference of statistical dependency networks therefore represents a form of large-scale hypothesis generation (Butte *et al.*, 2000; Opgen-Rhein and Strimmer, 2007a). It is then the task of the experimentalist to determine the biological causes that explain *why* the statistical dependencies exist. Where experimental evidence already exists, this is often used to validate (subnets of) the inferred network (e.g. Christley *et al.*, 2009).

Clearly, in order to construct a statistical dependency network, it is first necessary to define how the dependency will be assessed. Methods commonly employed in the literature include correlation (Butte *et al.*, 2000), partial correlation (Opgen-Rhein and Strimmer, 2007a), and mutual information (Margolin *et al.*, 2006). In this chapter, we focus upon partial correlation networks. We employ the `R` implementation of the GeneNet algorithm (Schäfer *et al.*, 2006; Opgen-Rhein and Strimmer, 2007a), and consider the *A. thaliana* data set of Smith *et al.* (2004) that is included with the package.

5.1.1 The data

The data set comprises time course measurements for 800 *Arabidopsis thaliana* (thale cress) genes. For each gene, there are 2 measurements at each of 11 different time points (representing 0, 1, 2, 4, 8, 12, 13, 14, 16, 20 and 24 hours from the start of the experiment). The aim of the original investigation was to analyse the changes in gene expression that occur during the diurnal cycle. To this end, the experiment involved changing the exposure of the plants to light over the course of a 24 hour period. The first measurement (at time 0) was taken immediately at the end of a 12 hour period of light. The next 5 measurements (at 1, 2, 4, 8 and 12 hours) were taken during a period of darkness. A 12 hour period of light followed, during which a further 5 measurements were taken (at 13, 14, 16, 20 and 24 hours after the start of the experiment). Each sample was obtained by harvesting three leaves from 20 plants. Microarray analyses were then performed upon each sample in order to quantify gene expression. The data were normalised, and then preprocessed to identify genes for which there was evidence of periodicity in the pattern of expression (see Smith *et al.*, 2004 for further experimental details, and Opgen-Rhein and Strimmer, 2007a; Wichert *et al.*, 2004 for preprocessing steps).

5.1.2 The GeneNet package

The GeneNet package allows partial correlation networks to be constructed from gene expression time course data. Given any collection of covariates, the partial correlation between any particular pair is (informally) the correlation that remains between them once the effects of all of the others have been regressed away. For the purposes of constructing networks, partial correlation is considered a far more informative measure of similarity

than simple correlation (Opgen-Rhein and Strimmer, 2007a). For example, if we have three correlated covariates, A, B and C, partial correlation allows us to quantify the degree to which the correlation between A and B is explained by the fact that both are correlated with C. If all of the covariates in our collection were jointly distributed according to a multivariate Gaussian distribution (in which case our inferred network would be a *graphical Gaussian model* — see Lauritzen, 1996), then this would be equivalent to determining the extent to which A and B are *conditionally independent* given C.

Partial correlations can be calculated by inversion of the usual correlation matrix (Schäfer and Strimmer, 2005). That is, denoting the correlation matrix by P, and the inverted correlation matrix by $P^{-1} = \Omega = (\omega_{ij})$, the partial correlation matrix $R = (r_{ij})$ is given by,

$$r_{ij} = -\frac{\omega_{ij}}{\sqrt{\omega_{ii}\omega_{jj}}}. \tag{5.1}$$

In order to make use of this formula, we clearly need to be able to calculate the correlation matrix. Given that we are interested in time courses of data, this means that an appropriate definition of correlation is required. GeneNet employs *dynamical correlation*, as defined in Opgen-Rhein and Strimmer (2006a). Briefly, the dynamical correlation between functions $g(t)$ and $f(t)$ is defined via the functional inner product,

$$\langle g(t), h(t) \rangle = \int_A^B g(t)h(t)dt, \tag{5.2}$$

where $[A, B]$ is the time interval of interest (here, A represents the first time at which an observation is made, and B the last). Time-centred functions, $g^C(t) = g(t) - \langle g(t), 1 \rangle$ and $h^C(t) = h(t) - \langle h(t), 1 \rangle$, may then be computed, after which variances may be defined as $\text{Var}(g(t)) = \langle g^C(t), g^C(t) \rangle$ (and similarly for h). This allows us to standardise the functions to obtain, for example, $g^S(t) = g^C(t)/\sqrt{\text{Var}(g(t))}$. Finally, the dynamical correlation is defined as,

$$\text{Cor}(g(t), h(t)) = \langle g^S(t), h^S(t) \rangle. \tag{5.3}$$

When we have only discrete, noisy observations of the functions g and h (which will always be the case in practice), the dynamical correlation must be estimated. We refer to Opgen-Rhein and Strimmer (2006a,b) for details of the procedure that is implemented in GeneNet.

Once we have computed the (dynamical) correlation matrix for our collection of time courses, we may use Equation (5.1) in order to estimate the partial correlation matrix, R. However, in order to proceed, we must then determine which of the partial correlations are significant. This allows us to go from a continuous partial correlation matrix to a binary adjacency matrix, and hence defines a network. In order to do this, GeneNet adopts the procedure of Efron (2004), which we now briefly outline. It is assumed that the observed partial correlations represent samples from a mixture model,

$$f(\hat{r}) = \eta_0 f_0(\hat{r}) + (1 - \eta_0) f_A(\hat{r}), \tag{5.4}$$

where $\hat{r}$ denotes the estimated partial correlation, so that $f(\hat{r})$ describes the distribution of estimated values across all possible edges. This captures the following ideas. We assume that there is some true, underlying (and unknown) partial correlation network, which we suppose to be sparse (i.e. there are very few edges relative to the total number of possible edges). Let us denote by C the set of all pairs of nodes (i.e. the set of all possible edges), and by C_A the set of all edges. Then $C_0 = C \backslash C_A$ represents the set of all non-edges. If we knew the network, then we could consider the distribution, $f_0(\hat{r})$, of partial correlation estimates amongst the elements of C_0, and the distribution, $f_A(\hat{r})$, of partial correlation estimates amongst the elements of C_A (the set of edges). The f_0 and f_A notation is designed to be deliberately suggestive of "null" and "alternative" distributions (respectively). The mixing parameter, η_0, describes the proportion of the elements of C that are in C_0, and hence represents the prior probability that any pair of nodes randomly selected from C is in C_0.

We now suppose that we are presented with a pair of nodes, $e = (\text{node}_i, \text{node}_j)$, for which we know the estimated partial correlation to be $\hat{r}$. Then Bayes rule tells us that the probability that $e \in C_0$ is given by,

$$\begin{aligned} p(e \in C_0 | \hat{r}) &= \frac{p(\hat{r} | e \in C_0) p(C_0)}{p(\hat{r})} \\ &= \frac{\eta_0 f_0(\hat{r})}{f(\hat{r})}. \end{aligned} \tag{5.5}$$

From this it follows that the posterior probability that e is a genuine edge is given by,

$$p(e \in C_1 | \hat{r}) = 1 - \frac{\eta_0 f_0(\hat{r})}{f(\hat{r})}. \tag{5.6}$$

In order to make a decision, we require a threshold, τ, so that if $p(e \in C_1 | \hat{r}) > \tau$ then we classify e as belonging to C_1 and hence determine that there should be an edge between node_i and node_j in our network.

It follows that, given our estimated partial correlation matrix, a value for τ, and Equation (5.4), we are able to classify all pairs of nodes as either edges or non-edges. Of course, in practice, we do not know f_0, f_A or η_0, and hence these must be estimated from the empirical distribution of partial correlation estimates. In GeneNet, this is accomplished by assuming a parametric form for f_0 (namely, the theoretical null for the sample normal partial correlation, as given in Hotelling, 1953; Schäfer and Strimmer, 2005), and estimating f_A nonparametrically from the empirical distribution (see Efron, 2004; Strimmer, 2008). The mixture model fitted to the *A. thaliana* data set is illustrated in Figure 5.1.

Once we have chosen a value for τ, the GeneNet algorithm may be treated as a black box, which returns the estimated partial correlation network corresponding to a particular input data set (as illustrated in Figure 5.2). It follows that, in order to assess the stability of the inferred networks, we may adopt a bootstrap procedure whereby we plug each replicate data set into the algorithm and assess the variability amongst the resulting networks.

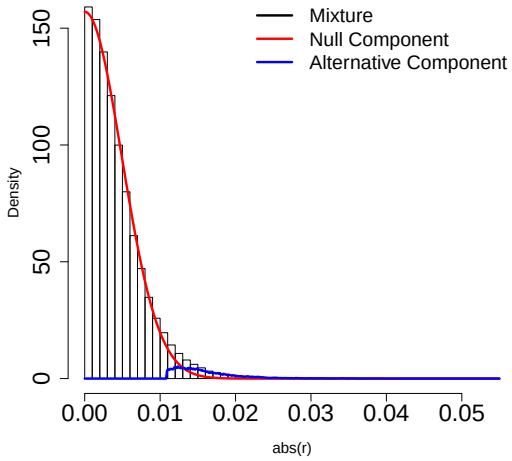


Figure 5.1: Illustration of the mixture model (Equation 5.4) fitted to the *A. thaliana* data set. The histogram (labelled "Mixture") shows the empirical distribution of the observed partial correlation estimates, which is taken as $f(\hat{r})$. The dotted red "null component" curve describes the fitted (parametric) null distribution, $f_0(\hat{r})$, while the blue curve describes the (nonparametric) alternative distribution, $f_A(\hat{r})$. Note that this figure is taken directly from the output of the GeneNet package.

5.2 Bootstrapping the data

As mentioned in Section 5.1.1, the *A. thaliana* data set comprises time courses for which there are two measurements at each of the time points. We here consider how we should treat data such as these, before describing the methods that we use to approximate the DGP.

5.2.1 Cross-sectional versus longitudinal data

We start by examining the different types of data that we might encounter. We can consider that there are broadly two sorts of time course data: *cross-sectional* and *longitudinal*. In addition to the exposition in this section, we would refer also to Storey *et al.* (2005), where different methods for coping with these two different types of data are discussed.

5.2.1.1 Cross-sectional data

These may be regarded as a collection of "snapshots" (cross-sections) taken at different time points (Figure 5.3a). Although we may have several measurements taken at each time point (possibly on different entities, or perhaps multiple noisy measurements on the same entity), we have no way to match up any *particular* measurement at time t_i with

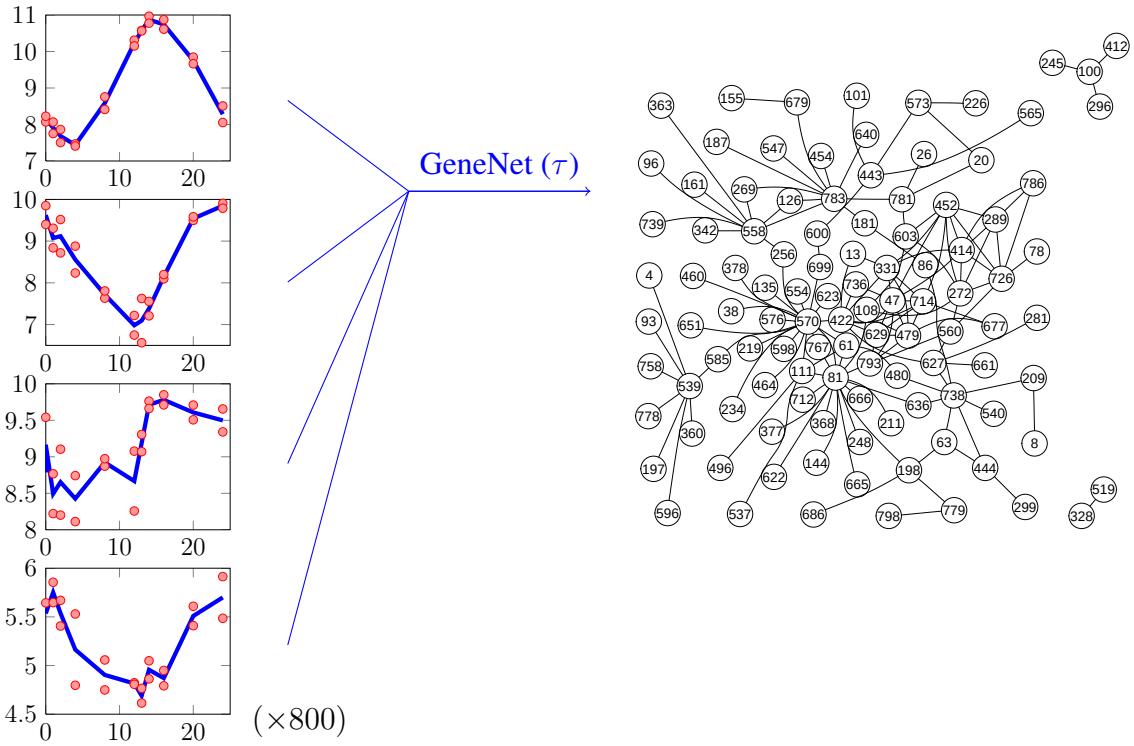


Figure 5.2: Illustration of the GeneNet algorithm, considered as a black box. In order to use the algorithm, we simply have to provide a data set (in our case, 800 gene expression time courses), and also the cutoff parameter, τ. The output is a partial correlation network. In the language of Chapter 1, we view the network as a *summary* of the observed data set, and use bootstrapping as a means to assess its stability.

measurements at any other time point t_j. These data are commonly encountered in the biosciences, where the process of taking measurements on a cell or organism is often extremely invasive and may preclude the possibility of taking measurements on the same entity at any later time point. In the extreme case, where the act of measurement destroys the sample or kills the organism, it is clear that each sample/organism may contribute measurements at only a single time point. Modelling such data involves trying to describe how a population of measurements evolves over time. One way of doing this is to assume a fixed parametric model for the population at every time point, so that our task is reduced to modelling how the parameters vary with time. Typically, we phrase such an approach as a regression problem. For example, we might assume that the measurements at time t may be modelled by a univariate normal distribution, $\mathcal{N}(f(t), \sigma^2)$, where σ^2 is fixed and $f(t)$ describes how the (true, unobserved) mean of the population varies over time. This is equivalent to the regression model,

$$y(t) = f(t) + \epsilon, \tag{5.7}$$

where $y(t)$ are the observed measurements and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. More complicated procedures might (for example) consider parametric models other than the normal distribution

and/or might allow the variance to change over time (e.g. Imoto *et al.*, 2003).

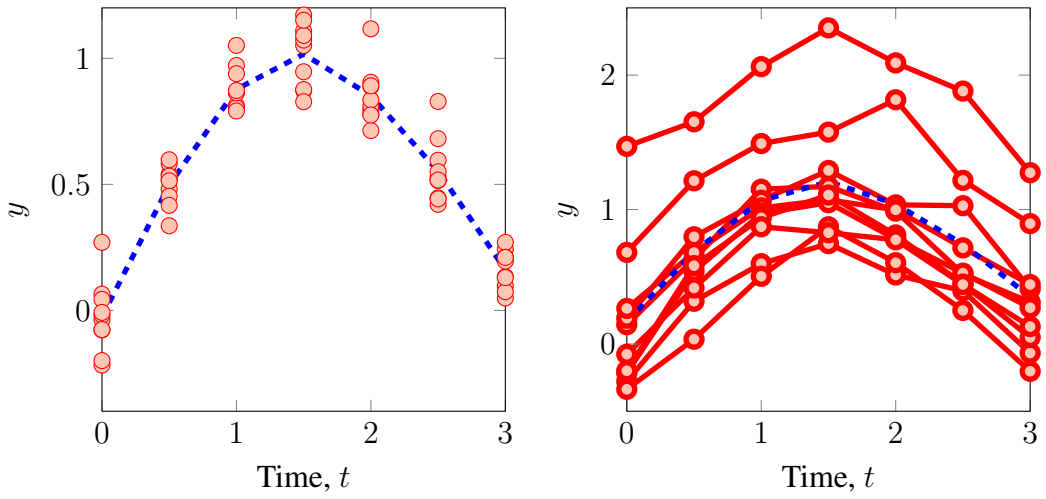


Figure 5.3: Cross-sectional versus longitudinal data. Left: synthetic cross-sectional data. Right: synthetic longitudinal data, where lines connect measurements taken on the same individual/entity. In both cases, the blue dashed line describes the empirical mean.

5.2.1.2 Longitudinal data

These comprise measurements on a single entity (or a collection of such entities) that is followed over time. For example, we may have a cohort of patients in whose response to a particular drug we are interested. A *longitudinal study* of these individuals would track each patient over the course of a period of time. For each individual, we would hence have measurements at several time points. Such data sets may be represented as in Figure 5.3b, with lines drawn between measurements taken on the same individual. In order to describe these data, we typically adopt a model similar to the one described by Equation (5.7), but with an additional index i to allow different models for different individuals. That is, we model the observed measurements, $y_i(t)$, taken on the i^{th} individual at time t as,

$$y_i(t) = f_i(t) + \epsilon_i, \tag{5.8}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$. Common simplifications include the assumptions that $\sigma_i = \sigma$ is the same for all individuals, and that we may write $f_i(t) = f(t) + \gamma_i$ for all i, so that the functions $f_i(t)$ are identical, up to an additive constant (see Storey *et al.*, 2005).

5.2.1.3 Practicalities

In practice, the distinction between longitudinal and cross-sectional data may not be as clear as suggested by the preceding paragraphs. In the case of the *A. thaliana* data set, for

example, the reason for which there are two measurements at each time point is because the entire experiment was repeated (Smith *et al.*, 2004). Thus, viewed at the "experiment level", we have two longitudinal time courses for each gene. That is, using the terminology of Section 5.2.1.2, we may regard the data as longitudinal if each experiment is treated as a different "individual". However, at the "plant level", the data are cross-sectional. Whether we should treat the data as longitudinal or cross-sectional depends upon our aims. When bootstrapping the data, we consider both possibilities, and assess the effects that this has upon our results.

5.2.2 Applying the bootstrap

We employ both the approach of Chapter 4 and the nonparametric bootstrap (Section 1.4.1.1), and consider cross-sectional and longitudinal cases separately (as described in Sections 5.2.2.1 and 5.2.2.2 below). In each case, we generate 1,000 replicate data sets. We fix τ and, for each replicate data set, apply the GeneNet algorithm in order to obtain a partial correlation network. Using the same value for τ, we also apply the algorithm to the original *A. thaliana* data set, D^{obs}.

5.2.2.1 Nonparametric bootstrap

Cross-sectional case Let y_{ij1} and y_{ij2} be the two measurements on gene i at time j. We obtain bootstrap replicates of y_{ijk} by drawing 2 samples with replacement from the set $\{y_{ij1}, y_{ij2}\}$. There are thus three possible outcomes: $\{y_{ij1}, y_{ij2}\}, \{y_{ij1}, y_{ij1}\}$ and $\{y_{ij2}, y_{ij2}\}$. Although this is relatively few, we repeat this for each of the 11 time points for gene i, and hence there are 3^{11} distinct possibilities for the bootstrapped time course. We repeat this process (independently) for each gene in order to obtain replicate data sets.

Longitudinal case Let $\mathbf{y}_{i1}$ and $\mathbf{y}_{i2}$ be the two vectors of measurements on gene i (so that the "1" and "2" label the two different experiments). We obtain bootstrap replicates of $\mathbf{y}_{ik}$ by drawing 2 samples with replacement from the set $\{\mathbf{y}_{i1}, \mathbf{y}_{i2}\}$. There are thus three possible outcomes: $\{\mathbf{y}_{i1}, \mathbf{y}_{i2}\}$, $\{\mathbf{y}_{i1}, \mathbf{y}_{i1}\}$ and $\{\mathbf{y}_{i2}, \mathbf{y}_{i2}\}$. For each gene, there are hence only 3 distinct bootstrapped time courses. However, because we test the interaction of each gene with 799 other genes, this actually amounts to a large number of distinct possibilities for the estimated partial correlation matrix, $\hat{R}$.

5.2.2.2 Multivariate Gaussian bootstrap

Cross-sectional case For gene i, we calculate the set of mean values $\{\overline{y}_{ij}\}_{j=1}^{11}$, where $\overline{y}_{ij}$ denotes the mean of y_{ij1} and y_{ij2}. We hence have a time course of mean values, to

which we can apply the approach of Chapter 4. It follows that, in this case, the "noise" variance, σ_ϵ^2, actually represents the variance of the sampling distribution of the sample mean at each time point. For a population distributed according to $\mathcal{N}(\mu, \sigma^2)$, the sampling distribution of the sample mean is $\mathcal{N}(\mu, \sigma^2/n)$, where n is the number of observations used to calculate the sample mean (see, for example Freund, 1971). It follows that, in order to generate replicate observations (rather than generating replicates of the sample mean) , we have to modify Equation (4.20) to become,

$$\mathbf{y}^{\text{rep}} \sim \mathcal{N}(\mathbf{m}_n, K_n + n\sigma_\epsilon^2 I), \tag{5.9}$$

where, in our case, $n = 2$. We use this procedure in order to obtain replicate time courses for each gene independently. We take our covariance function, k, to be $k(t_i, t_j) = k_{SE}(t_i, t_j) + k_M(t_i, t_j)$, where k_{SE} is the squared exponential covariance function of Equation (4.11), and k_M is a *Matérn* covariance function,

$$k_M(t_i, t_j) = \sigma_f^2 \left(1 + \frac{\sqrt{3}|t_i - t_j|}{l_2}\right) \exp\left(-\frac{\sqrt{3}|t_i - t_j|}{l_2}\right), \tag{5.10}$$

where σ_f and l_2 are hyperparameters that we estimate as previously. The Matérn class of covariance functions is recommended by some authors (Stein, 1999), who feel that the strong smoothness properties of the squared exponential covariance function are unrealistic (see Rasmussen and Williams, 2005, for a discussion of the properties of various covariance functions). We here use a combination of squared exponential and Matérn covariance functions, as this allows for greater flexibility in the range of behaviours that may be modelled.

Longitudinal case For gene i, we simply apply the approach of Chapter 4 to the two time courses, $\mathbf{y}_{i1}$ and $\mathbf{y}_{i2}$, independently. We use the same covariance function as in the cross-sectional case.

5.3 Results

5.3.1 Illustration with $\tau = 0.999999$

Initially, we take τ to be very close to 1 (namely, $\tau = 0.999999$). This represents very stringent control over the so-called *local* false discovery rate (Efron *et al.*, 2004), since we require a putative edge to have a very high posterior probability in order for us to accept it as genuine. In practice, we would be very unlikely to set such a stringent threshold (Efron *et al.*, 2004 and Schäfer *et al.*, 2006 recommend a value of around 0.8). However, it here proves a very effective way in which to simplify the presentation of our results, as the resulting networks have relatively few edges and are therefore much easier to visualise

than the oft-criticised "hairball" (or "ridiculogram") images that pervade the biological literature (Lander, 2010).

In Figure 5.4, we show the network inferred from D^{obs}, in which the node labels correspond to the gene identifiers used within GeneNet (see Appendix A for a table that allows these labels to be decoded). The edges are coloured according to their *bootstrap support* (here calculated using the longitudinal nonparametric bootstrap procedure described in Section 5.2.2.1). The bootstrap support for an edge is simply the proportion of the bootstrapped networks that also contained that edge. Bootstrap support is often considered when estimating phylogenies (e.g. Brown, 1994; Felsenstein, 1985), and is here used as an assessment of edge stability (very similar to the use of estimated selection probabilities in Chapter 2). Note that the colour of the nodes is a visual aid only, to help us to locate edges that have a high degree of bootstrap support. Specifically, each node is given the same colour as whichever of its incident edges has the highest bootstrap support.

In Figure 5.5, we show similar plots obtained using each of the bootstrap procedures described in Section 5.2.2.

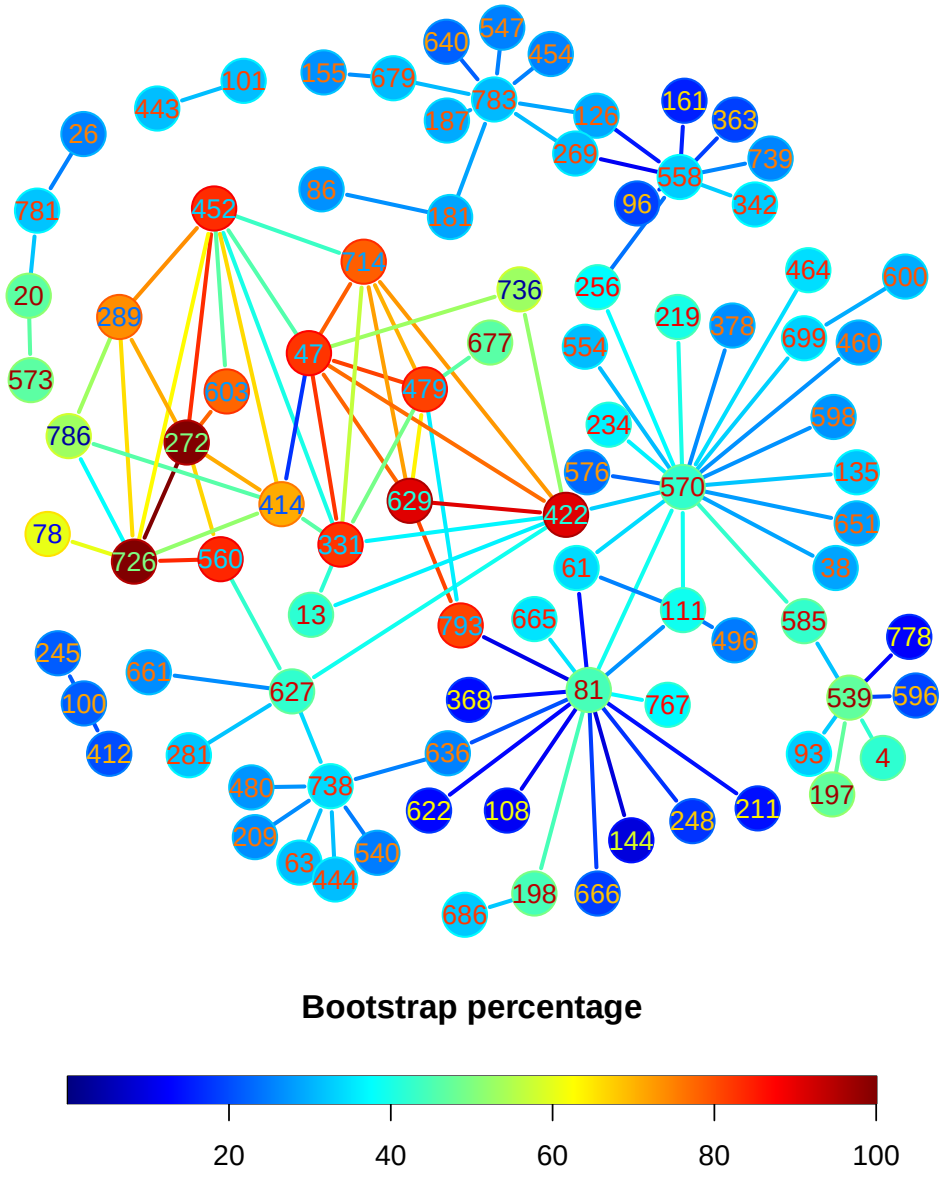
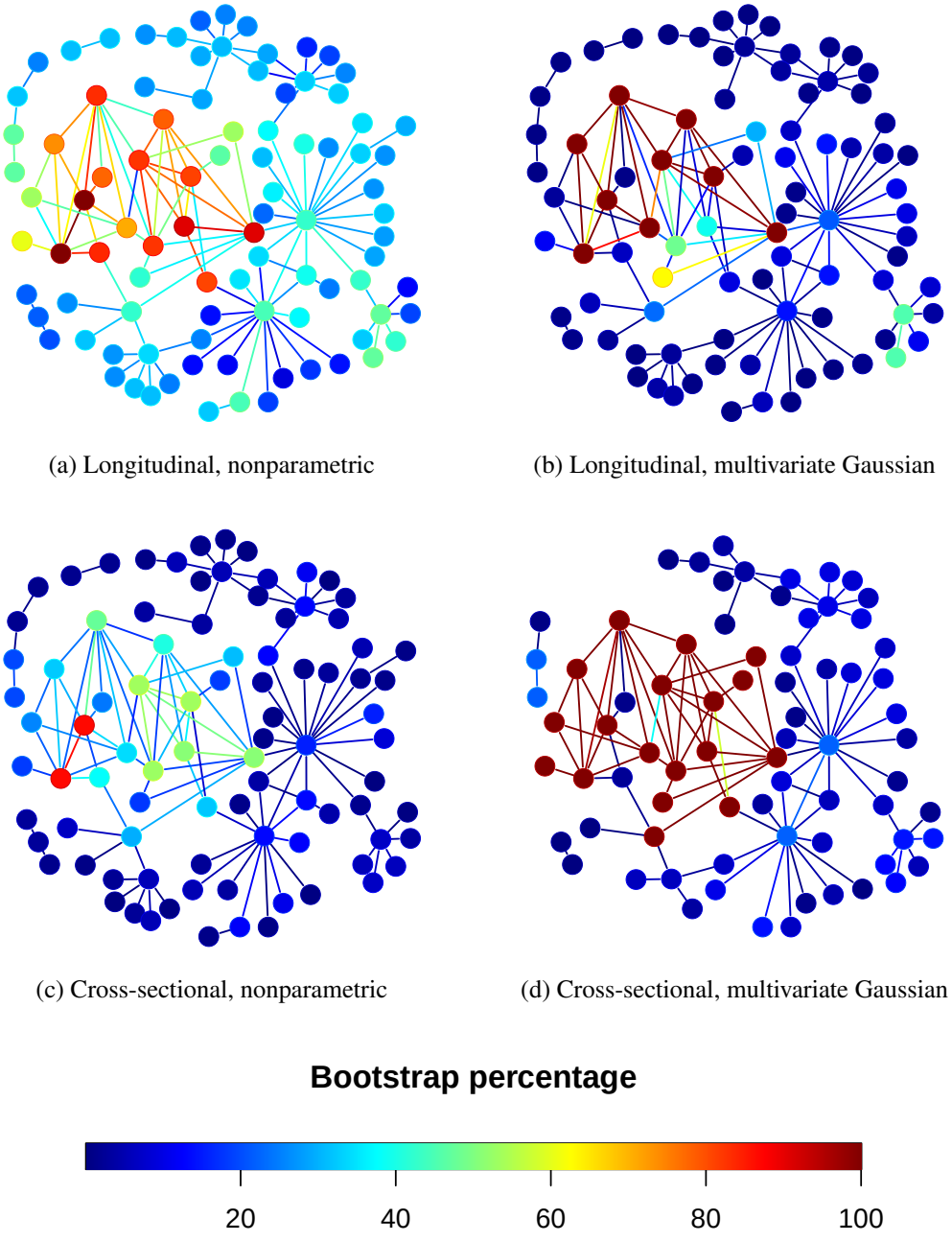


Figure 5.4: Network inferred from the original *A. thaliana* data set, when $\tau = 0.999999$. Edges are coloured according to their bootstrap support, with reds indicating higher and blues indicating lower support, as indicated in the colour bar. Each node is given the same colour as whichever of its incident edges has the highest bootstrap support. Labels on the nodes allow the corresponding genes to be identified (see Appendix A). All orphan nodes are omitted.



(a) Longitudinal, nonparametric

(b) Longitudinal, multivariate Gaussian

(c) Cross-sectional, nonparametric

(d) Cross-sectional, multivariate Gaussian

Figure 5.5: As in Figure 5.4, but considering all 4 procedures for bootstrapping the data. Node labels are omitted, as are any edges for which the bootstrap support is 0%. (a) Using the nonparametric bootstrap when the data are treated as longitudinal (this is identical to Figure 5.4); (b) Using the multivariate Gaussian bootstrap of Chapter 4 and again treating the data as longitudinal; (c) Nonparametric bootstrap, data treated as cross-sectional; (d) Multivariate Gaussian bootstrap, data treated as cross-sectional.

Although there are differences between the bootstrap support percentages that arise using the different methods, there is a general pattern that is consistent through Figures 5.5a – 5.5d. We can see that there is a highly connected region of the network (which includes, amongst others, the nodes labelled as 726, 272 and 47 in Figure 5.4) where there are many edges with a high level of bootstrap support. Outside of this region, edges have a low level of bootstrap support.

The bootstrap support percentages calculated using the "longitudinal, nonparametric" bootstrap were generally higher than for the other 3 bootstrap approaches (as can be seen immediately from the much "brighter" appearance of Figure 5.5a relative to Figures 5.5b – 5.5d). This is what we would expect, as the bootstrap approach used to produce Figure 5.5a is the most constrained (as we discussed in Section 5.2.2.1, for each gene, one in three of the replicate data sets will be identical to the original). Indeed, given the constrained nature of this approach, it is quite surprising that so many of the edges have such low bootstrap support (in the blue range of the colour bar). In Section 5.3.2 we investigate whether or not this is connected to our (unrealistic) choice of τ .

5.3.2 A range of values for τ

The value for τ that we considered in the previous section was much higher than we would take in practice (Efron *et al.*, 2004, suggests a value of around 0.8). We here consider a range of values, $\tau = 0.45, 0.55, \ldots, 0.95$. Unfortunately, for these smaller cutoff values, the network becomes much harder to visualise (the network inferred from D^{obs} has 1,917 edges when $\tau = 0.95$ and 10,467 edges when $\tau = 0.45$). Instead, we assess the agreement between our bootstrap replicate networks and the original networks using a similarity measure of the type discussed in Chapter 2. Let $E^{\text{obs}}(\tau)$ denote the set of all edges that appear in the network inferred from D^{obs} when the cutoff is set to τ. Similarly, let $E_i^{\text{rep}}(\tau)$ denote the set of all edges that appear in the network inferred from the i^{th} replicate data set, D_i^{rep} when the cutoff is set to τ. We define the *proportion of overlapping edges* to be,

$$\rho_i(\tau) = \frac{|E_i^{\text{rep}}(\tau) \cap E^{\text{obs}}(\tau)|}{|E^{\text{obs}}(\tau)|}. \tag{5.11}$$

We note that this similarity measure is identical to the "percentage of overlapping genes" or "percentage of overlapping features" (Shi *et al.*, 2005; He and Yu, 2010), but used in a different context. We calculate $\rho_i(\tau)$ for each of our replicate data sets, and hence obtain a bootstrap distribution for $\rho(\tau)$. Given that we have four different bootstrap procedures (Section 5.2.2), we have four bootstrap distributions for each value of τ. The resulting histograms are provided in Figure 5.6.

Figure 5.6 shows that the networks inferred from the replicate data sets generated using from the longitudinal, nonparametric bootstrap procedure are most similar to the original network, with $\rho_i(\tau)$ values generally falling between 0.3-0.4. This chimes with our intuition regarding the constrained nature of this procedure. The other bootstrap procedures

give rise to broadly similar distributions, with $\rho_i(\tau)$ values generally falling between 0.1-0.2. Within the range considered, the precise value of τ makes little difference to the bootstrap distribution of $\rho(\tau)$. Although the numerator in Equation 5.11 increases with τ, the denominator also does so (and at a similar rate).

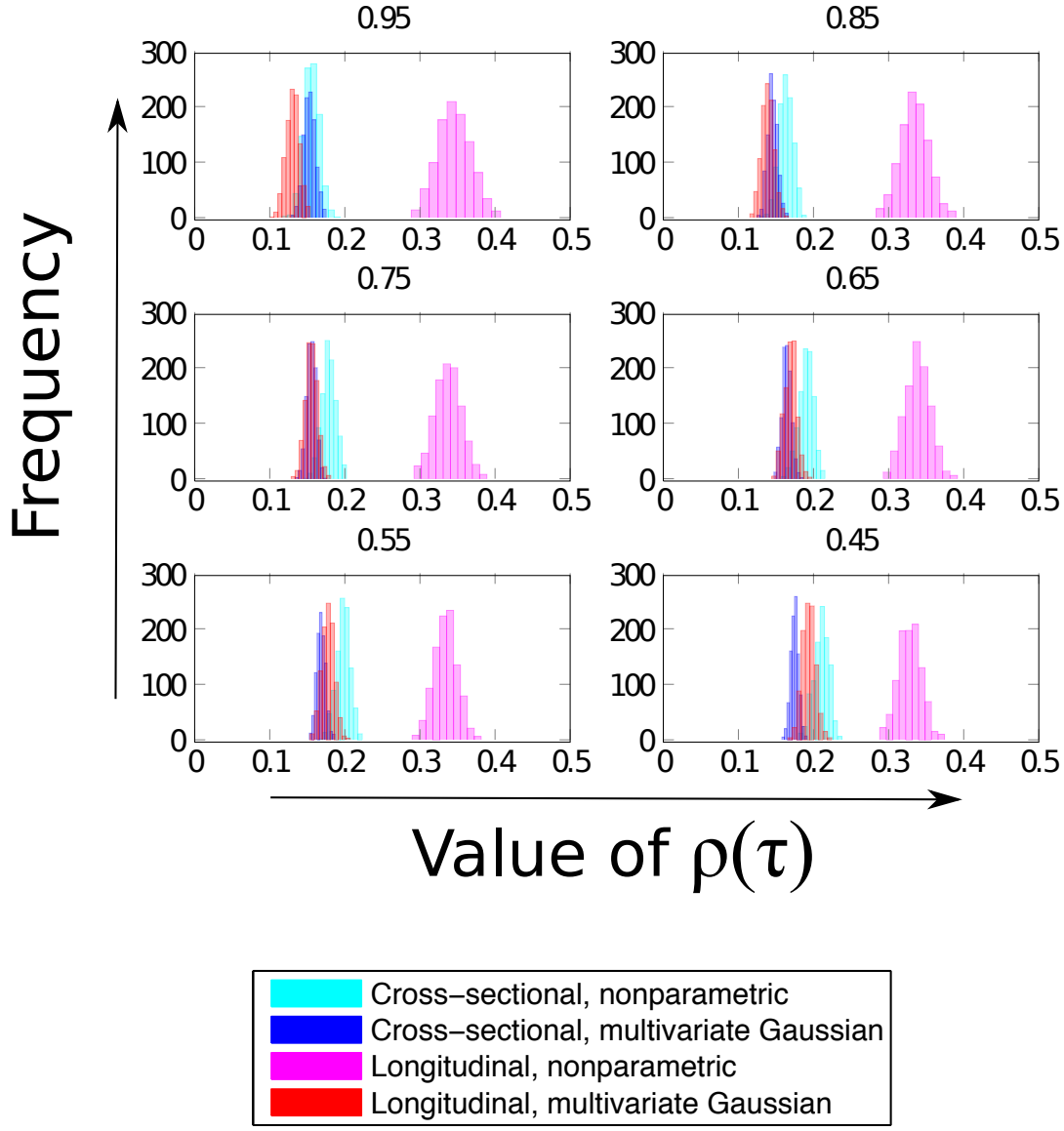


Figure 5.6: Plots showing the bootstrap distributions of $\rho^{(i)}(\tau)$ for different values of τ and four different bootstrap approaches. The title above each histograms indicates the value of τ, while the colours of the bars indicate the bootstrap procedure.

It is again striking that there are relatively few stable edges (at best, only around 30-40% of the edges in the original network appear in the replicate networks). We discuss possible reasons for this in Section 5.4.

5.3.3 Stable subnets

Thus far, our results have been rather negative, suggesting that the estimated network is relatively unstable to realistic perturbations of the data. We now consider how the bootstrap replicates may be used more positively. Motivated by Chapter 2, we identify the most "stably selected" edges and hence generate stable subnets of the original network. To do this, we simply specify a bootstrap support threshold, b, and then omit the edges in the network that do not have at least this level of bootstrap support. In a different context (determining whether or not a set of species forms a monophyletic group), $b = 95\%$ is commonly considered to represent a "significant" level of bootstrap support (Felsenstein, 1985), although this is considered by some to be overly conservative (e.g. Brown, 1994). We here do not attempt to determine the "correct" choice of b (although we note that this problem is closely related to the problem of choosing π_{thr} in Section 2.4, and could provide an interesting direction for future study). Instead, we simply note that the bootstrap support provides us with a way of assigning stability scores to the edges, and hence could be used as part of a "stability selection" approach for edges. A basic example is given in Figure 5.7.

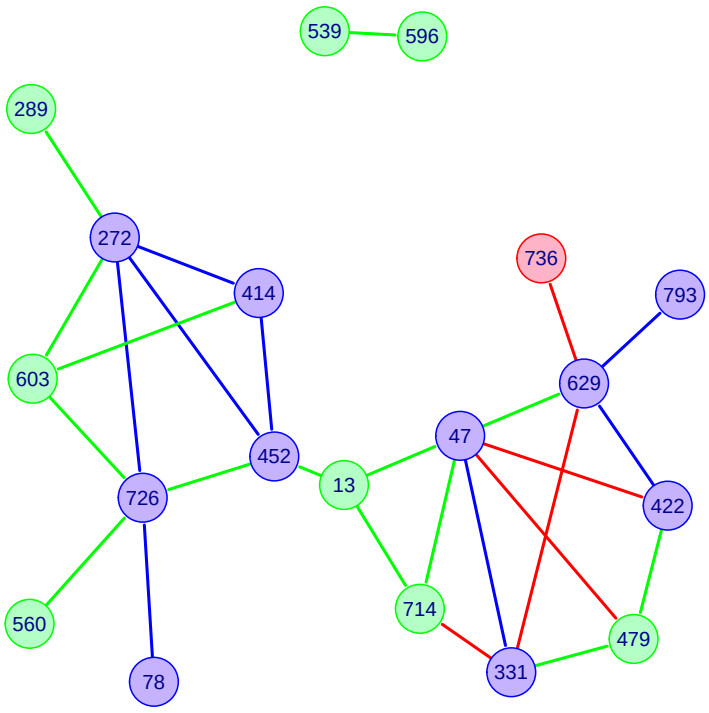


Figure 5.7: A "stable subnet" of the network estimated from D^{obs} when $\tau = 0.99$. Edges with 100% bootstrap support are coloured blue; those with between 95 and 100% are coloured green; and those with between 90 and 95% are coloured red. All other edges are omitted. Each node has the same colour as whichever of its incident edges has the highest bootstrap support. Bootstrap support was calculated using the longitudinal, nonparametric approach.

5.4 Discussion

We applied the multivariate Gaussian bootstrap approach of Chapter 4, as well as nonparametric procedures, in the context of gene network inference. We considered only the GeneNet algorithm of Schäfer *et al.* (2006), although we note that there are now several further methods for gene network inference which could also be investigated using the approaches described (e.g. Lèbre, 2009; Opgen-Rhein and Strimmer, 2007b). Our bootstrap approaches were adapted to reflect different interpretations of the data as longitudinal or cross-sectional. However, regardless of the approach we considered, we found the inferred network to be relatively unstable.

In order to gain a little more insight into the apparent instability of the inferred network, we now briefly consider the bootstrap distributions for the estimated partial correlations. We focus on the partial correlation between the genes labelled as 726 and 272 in Figure 5.4, and also between the genes labelled as 81 and 144. We choose these particular nodes, because a very stable edge was consistently found between the first pair of genes (for all values of τ), while the bootstrap support for the edge between 81 and 144 was low (for $\tau = 0.999999$). Using our longitudinal, nonparametric bootstrap replicates, we obtain bootstrap distributions for these partial correlations. These are shown as histograms[1] in Figure 5.8, overlaid on top of the fitted mixture model of Figure 5.1 (in order to provide a sense of scale).

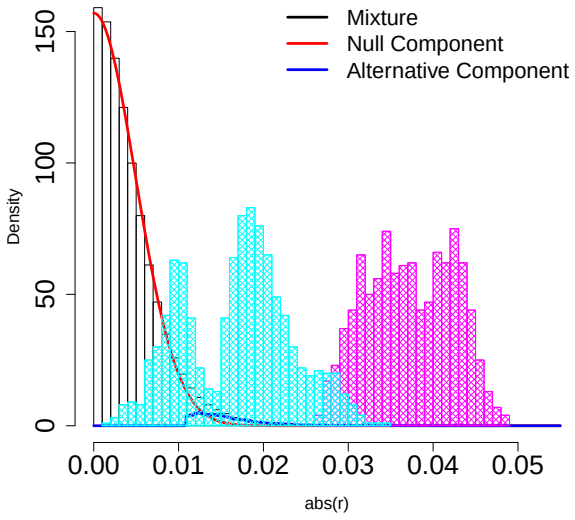


Figure 5.8: Bootstrap distributions of (absolute) estimated partial correlations for: genes 726 and 272 (magenta); and genes 81 and 144 (cyan). These are plotted over the fitted mixture model of Figure 5.1, in order to provide a sense of scale.

[1]For consistency with Figure 5.1, we actually plot the absolute values of the estimated partial correlations.

We can see that the magnitude of the partial correlation between genes 726 and 272 is consistently large across all bootstrap replicates, and hence it is unsurprising that it is stably determined to correspond to an edge. In contrast, the magnitude of the partial correlation between genes 81 and 144 varies between 0.0018 (very low) and 0.034 (high). It is therefore unsurprising that, while we occasionally determine there to be an edge between these two genes, we often conclude the opposite. However, this does not explain *why* the partial correlation between genes 81 and 144 varies between such a low value and such a high value. Our current hypothesis is that this is (at least in part) due to difficulties caused by the presence of multiple strongly correlated genes (in a manner that is closely related to the problems discussed in Chapter 2, Section 2.4.2). Recall our earlier example in which we considered three correlated covariates (A, B, C). We stated that partial correlation allows us to quantify the degree to which the correlation between A and B is explained by the fact that both are correlated with C. Now, in the original data set, there are 242 genes whose dynamical correlation with both gene 81 *and* gene 144 is stronger (larger in magnitude) than the dynamical correlation of these two genes with each other. It follows that, in order for there to be a stable edge between genes 81 and 144, we would consistently have to determine that the correlation between these two genes is not explained by any of the other 242 genes. Given that our bootstrap procedures perturb the time courses for *all* of these genes (independently), it seems unsurprising that we are unable to do this. For completeness, we note that there are no genes whose dynamical correlation with gene 726 and gene 272 is stronger than their dynamical correlation with each other.

At this stage, the above is just a hypothesis. We would suggest that in order to investigate this further, it would be prudent to undertake a series of simulation studies (similar to those of Chapter 2) in which the dynamical correlation between time courses is controlled. Similar studies have been performed in order to investigate the quality of networks inferred from *static* data (Werhli *et al.*, 2006), but we are unaware of any examples where GeneNet has been applied to simulated time course data. We do not pursue this further in the current thesis, but we believe that this presents an important direction for future work.

Chapter 6

Application II: ODE parameter estimation

Overview

Abstract *We apply the bootstrap of Chapter 4 in the context of estimating parameters of ordinary differential equation (ODE) models. We illustrate using a simulated Lotka-Volterra model, and also consider the estimation of parameters from real experimental data in an ODE model of the JAK2-STAT5 signalling pathway. We briefly discuss how the Gaussian process regression model used to perform the bootstrap can also be used as part of the estimation procedure itself.*

Overview *We start in Section 6.1 by providing some brief background regarding ODE modelling in systems biology, focusing on methods for parameter estimation. In Section 6.2, we consider a simulation example, using artificial data generated from a Lotka-Volterra model. We move on to a real example in Section 6.3, where we estimate the parameters of an ODE model of the JAK2-STAT5 cell signalling pathway. In Section 6.4, we briefly discuss a possible extension of our approach, which we demonstrate can speed-up the process of parameter estimation. We end in Section 6.5 with a discussion of our results and conclusions.*

6.1 Background

Ordinary differential equation (ODE) systems have become widely used in systems biology, to model gene regulatory networks, signalling pathways and metabolic networks (e.g. Quach *et al.*, 2007; Swameye *et al.*, 2003; Tyson *et al.*, 2003; Schoeberl *et al.*, 2002). We

may consider the process of ODE modelling (and mechanistic modelling more generally) as comprising three steps: formulation, fitting, and prediction.

1. *Formulating* the model provides us with a formal means by which to express our current understanding of the biological system;
2. *Fitting* to an observed data set allows us to determine how well our understanding agrees with empirical evidence;
3. *Predicting* using the fitted model allows us to generate new, testable hypotheses (which may well lead to refinement of the original model).

We are concerned with the second of these steps. In general, an ODE model will have a number of unknown parameters (which may include initial conditions), and it is these that we must estimate in order to fit to the observed data. We wish to use the methods of Chapter 4 in order to quantify the stability of these estimates.

In this section, we provide a brief introduction to ODE modelling, and describe two broad methodologies for parameter estimation: simulation and two-step methods. In Sections 6.2 and 6.3 we consider examples that employ simulation based parameter estimation routines. However, our bootstrap approach lends itself naturally to a two-step method, and we discuss this further in Section 6.4.

6.1.1 Ordinary differential equation models

We consider ODE models of the form,

$$\frac{d}{dt}\mathbf{x}(t) = h(\mathbf{x}; \boldsymbol{\theta}). \tag{6.1}$$

Here, $\mathbf{x} \in \mathbb{R}^p$ is the vector of *state* or *phase* variables, and $\boldsymbol{\theta}$ is the vector of parameters. We will be concerned with *initial value problems*, where we know (or must estimate) the initial conditions $\mathbf{x}(0) = \mathbf{x}_0$. It will not usually be possible to solve Equation 6.1 analytically, and hence numerical methods must be employed. There is a wide variety of methods for solving systems of ODEs (see, for example, Press *et al.*, 2007). Throughout this chapter, we use an explicit Runge-Kutta (4,5) method (Dormand and Prince, 1980).

6.1.2 Parameter estimation for ODE models

There have been many methods proposed for estimating the parameters of ODE models of biochemical processes (for example Perkins *et al.*, 2006; Quach *et al.*, 2007; Toni *et al.*, 2009). We here classify all such methods as either *simulation* or non-simulation based methods. Amongst the latter, *two-step* approaches (discussed in Section 6.1.2.2) are perhaps the most common (Brunel, 2008)

6.1.2.1 Simulation methods

At their most basic, simulation methods may be considered as a form of trial and improvement approach. We start by choosing some initial value for the parameters, then solve the resulting system of ODEs and assess how close we are to the observed data. After this, we propose a new value for the parameters, solve the system again, and determine if we have a better or worse fit to the data. We repeat this process many times until either we are satisfied that the discrepancy between the fit and the data is small enough, or we are unable to make significant further improvements, or we reach a predefined maximum number of iterations. It is clear that such approaches require three main ingredients: (i) an *error function* to minimise; (ii) an *optimisation routine* to perform the minimisation; and, (iii) a *stopping criterion*. These are common requirements for optimisation problems, and we refer to Press *et al.* (2007) for descriptions of many suitable algorithms. An early case study of ODE parameter estimation using optimisation routines is provided by Biegler *et al.* (1986). Here it is noted that, instead of an error function, we might instead be able to define a *likelihood function*. For example, we might assume that our ODE model defines the mean behaviour of the system, and that the observed data represent noisy realisations of the system. If we assume a parametric form for the noise model (such as univariate Gaussian noise), then it is relatively straightforward to define a likelihood function (Biegler *et al.*, 1986; Kirk *et al.*, 2008). Once we have a likelihood function, we may use an optimisation procedure to find the maximum likelihood parameter values, or may adopt a Bayesian approach and seek the posterior distribution of the parameters given the data (Gelman *et al.*, 1996).

The difficulty with these simulation-based approaches is that, for each new parameter proposal, we are required to solve the ODE system. Depending on the optimisation routine and the nature of the error (or likelihood) surface, we might require tens or hundreds of thousands of simulations (or more) in order to obtain an adequate fit (Toni *et al.*, 2009). This can come at a huge computational cost, and hence simulation methods are often relatively slow.

6.1.2.2 Two-step methods

To avoid having to solve the ODE system, we might consider a *two step* approach, in which we first approximate $\mathbf{x}(t)$ by $\hat{\mathbf{x}}(t)$, and then find the $\boldsymbol{\theta}$ that minimises the discrepancy between $\frac{d}{dt}\hat{\mathbf{x}}(t)$ and $h(\hat{\mathbf{x}}; \boldsymbol{\theta})$ (Brunel, 2008). Although this again requires the use of an optimisation routine, we avoid the computational expense associated with having to solve the ODE system.

Varah (1982) provides an early example of the application of two-step methods in which the approximation $\hat{\mathbf{x}}(t)$ is found by modelling the observed data using splines. Different methods for approximating $\hat{\mathbf{x}}(t)$ include local polynomial regression (Jost and Ellner, 2000) and artificial neural networks (Voit and Almeida, 2004). One potential difficulty

with these methods is that they rely (to some degree) on the quality of the approximation $\hat{\mathbf{x}}(t)$. Even if $\hat{\mathbf{x}}(t)$ provides a good fit to the observed data, there is no guarantee that it matches the behaviour permitted by the ODE model. To address these difficulties, Poyton *et al.* (2006) proposed a procedure to estimate $\hat{\mathbf{x}}$ and $\boldsymbol{\theta}$ in an iterative fashion, which was later extended in Ramsay *et al.* (2007).

6.2 Example I: Lotka-Volterra model

We start by considering a simulation example, based upon a Lotka-Volterra model (Lotka, 1920; Volterra, 1926). Such models have been employed as a means to describe the relationship between predator and prey species in ecology, and also to describe interactions between "predator" and "prey" molecular species (Wilkinson, 2006).

6.2.1 The model

We consider the following version of the Lotka Volterra predator-prey model

$$\begin{aligned} \frac{dx}{dt} &= ax - xy \\ \frac{dy}{dt} &= bxy - y \end{aligned} \tag{6.2}$$

where a and b are parameters, and we assume that the initial conditions are known and gven by $x(0) = 1, y(0) = 0.5$. Here, x and y refer to the population sizes of the prey and predator species (respectively).

6.2.2 The data

We consider the artificial data set of Toni *et al.* (2009). These data were simulated by setting $a = b = 1$ in Equation (6.2), solving the equations numerically, and then adding Gaussian noise. Data were sampled at 8 time points (for both x and y), and the noise was generated by sampling independent random variates from a $\mathcal{N}(0, 0.5^2)$ distribution. We denote the data by $D^{\text{obs}} = \{(x_i, y_i)\}_{i=1}^{8}$, where (x_i, y_i) are the observed values for x and y at time t_i. These simulated data are shown in Figure 6.1, together with the solutions for $x(t)$ and $y(t)$ when $a = b = 1$ (labelled as "true x" and "true y"). The full data set is provided in Toni *et al.* (2009, Supplementary Material).

6.2.3 Estimating the parameters

We employ a simulation approach in order to perform least squares estimation of the parameters. That is, we seek the parameters, $\boldsymbol{\theta} = [a, b]$, that minimise the objective function,

$$g(\boldsymbol{\theta}; D^{\text{obs}}) = \sum_{i=1}^{8} \left((x(t_i; \boldsymbol{\theta}) - x_i)^2 + (y(t_i; \boldsymbol{\theta}) - y_i)^2 \right), \tag{6.3}$$

where $x(t_i; \boldsymbol{\theta})$ is the value of $x(t)$ at time $t = t_i$ when the parameters in Equation (6.2) are given by $\boldsymbol{\theta}$ (and similarly for $y(t_i; \boldsymbol{\theta})$).

In order to minimise $g(\boldsymbol{\theta}; D^{\text{obs}})$, we use the `lsqnonlin` function in Matlab (MATLAB, 2009), which employs a trust-region-reflective algorithm based on an interior-reflective Newton method (Coleman and Li, 1994, 1996). This allows us to specify bounds on the acceptable range of values for a and b, which we set to be $0 \leq a, b \leq 10$. We set the initial point for the optimisation algorithm to be $a = b = 0.5$. Since `lsqnonlin` finds local minima only, we should ideally perform several random initialisations (i.e. with different initial points) to try to avoid getting stuck in a local minimum. However, since this example is for illustration only, we use just one initial point in order to reduce computational overheads. As we shall see later, local minima *do* appear to be an issue for this example.

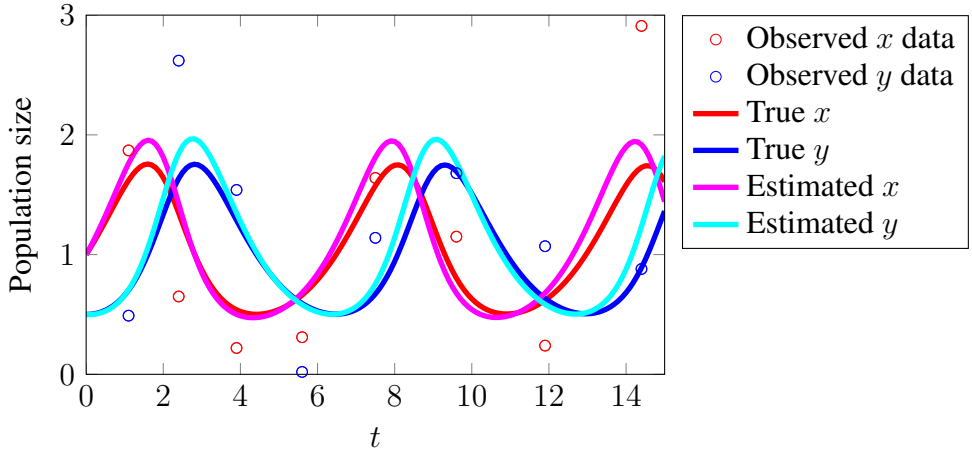


Figure 6.1: Lotka-Volterra model and simulated data. "True x" and "true y" show the model from which the data were simulated, while "estimated x" and "estimated y" are the fits provided by parameters estimated from the simulated data.

Before applying our bootstrap procedure, we use the `lsqnonlin` function to estimate the parameters from the original data set. We obtain $\hat{a}^{\text{orig}} = 1.0712$ and $\hat{b}^{\text{orig}} = 0.9585$. The fits provided by these parameters are shown in Figure 6.1 (labelled as "estimated x" and "estimated y"). In order to provide some context for our later analysis, we also consider a grid of a and b values, and for each pair we calculate $g(\boldsymbol{\theta}; D^{\text{obs}})$. The resulting error surface is illustrated in Figure 6.2, with the estimated parameter values, $(\hat{a}, \hat{b})$, indicated by a "+" symbol.

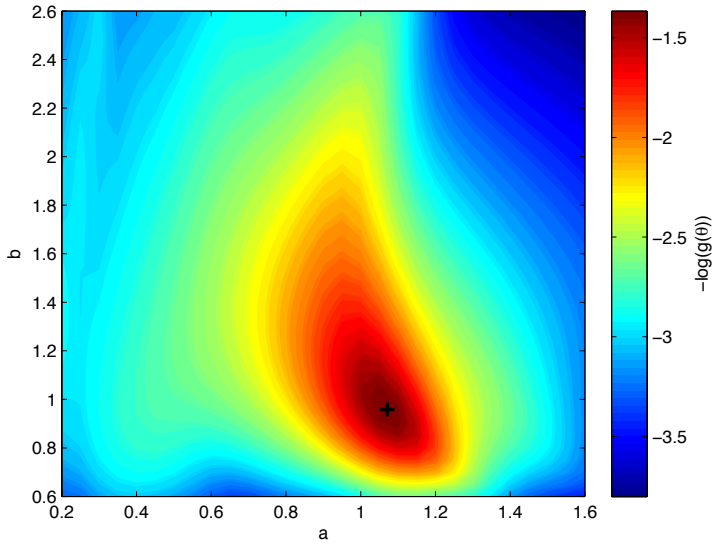


Figure 6.2: Representation of the negative log error surface for the Lotka-Volterra system, given data set D^{obs}. The black "+" sign marks the parameter values estimated from the original data set.

6.2.4 Bootstrap procedure

We fit Gaussian process regression models independently to the x data and to the y data. We use a zero mean function, and a squared exponential covariance function whose parameters are estimated by maximisation of the marginal likelihood. The resulting fits are shown in Figure 6.3. We sample 10,000 replicate data sets from the model, according to the procedure given in Chapter 4. For each replicate data set, D_i^{rep}, we fit the ODE model by using the `lsqnonlin` function to find the parameters that minimise $g(\boldsymbol{\theta}; D_i^{\text{rep}})$.

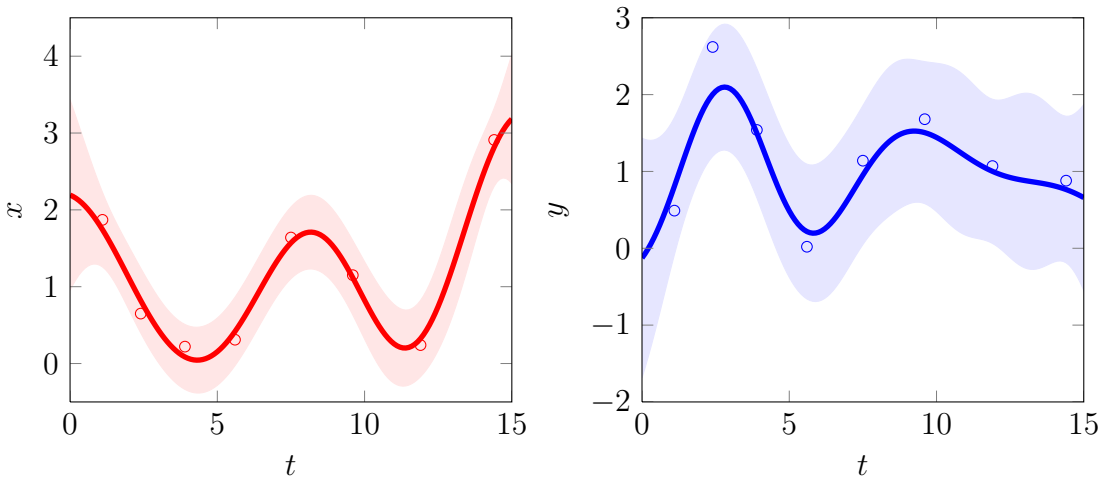


Figure 6.3: GP regression models fitted to the data for the x (left) and y (right) variables.

6.2.5 Results

Corresponding to each of our replicate data sets, D_i^{rep}, we have a parameter estimate, $\boldsymbol{\theta}_i^{\text{rep}} = [a_i^{\text{rep}} b_i^{\text{rep}}]$. Figure 6.4 illustrates the joint distribution of these estimates, while Figure 6.5 shows the marginal bootstrap distributions.

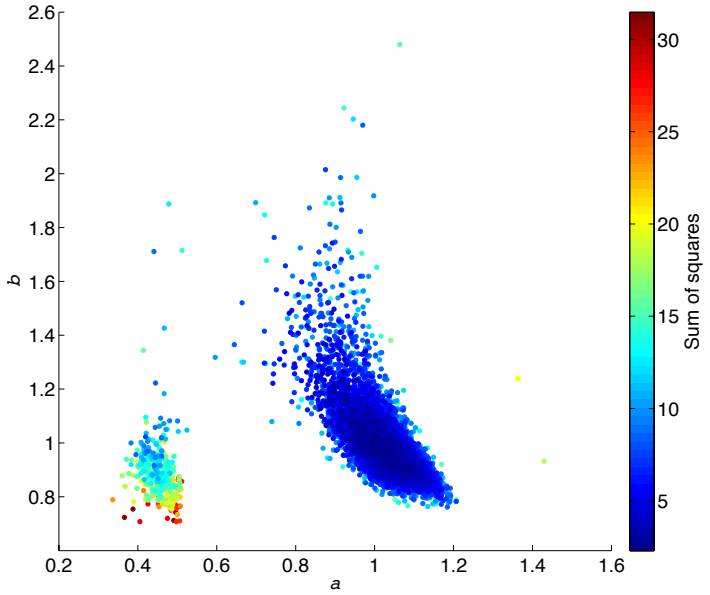


Figure 6.4: Scatterplot of the bootstrap distribution of estimated parameter values for the Lotka-Volterra example. Each point corresponds to a different $\boldsymbol{\theta}_i^{\text{rep}}$. The colours of the points indicate the value of $g(\boldsymbol{\theta}_i^{\text{rep}}; D_i^{\text{rep}})$, with blue corresponding to lower errors and red corresponding to higher errors.

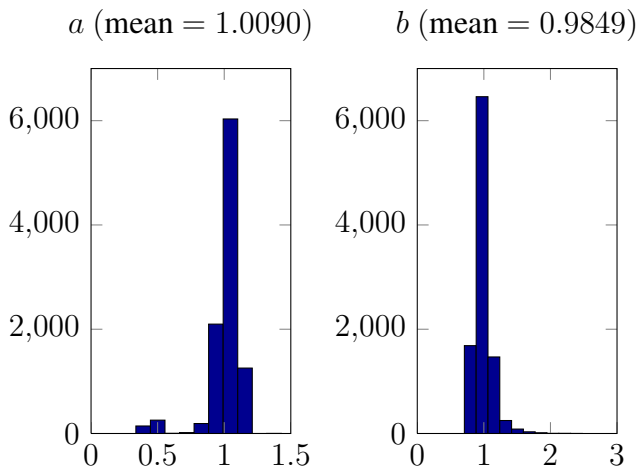


Figure 6.5: Histograms of the marginal bootstrap distributions of the estimates for a (left) and b (right). The mean values are provided above each plot.

Figures 6.4 and 6.5 both show that the majority of the estimated parameters are clustered around the true values, $a = b = 1$, with the large cluster in Figure 6.4 comprising approximately 96% of the points. This is reflected in the mean values of the estimates, which are 1.0090 and 0.9849 for a and b respectively. The estimated standard error for both a and b is 0.13 (to 2 significant figures). This is reassuring: it suggests that realistic changes to the observed data generally make quite small differences to our parameter estimates. However, it is notable that, for around 4% of our parameter estimates, we obtain quite different estimates, with $\hat{a} \approx 0.45$ and $\hat{b} \approx 0.8$. However, the fits provided by these estimates are generally quite a lot worse than the fits provided by those that fall in the large cluster in Figure 6.4. This suggests that these estimates might arise from a failure by the optimisation algorithm to converge to the true minimum.

If we overlay the scatterplot of Figure 6.4 onto the depiction given in Figure 6.2 of the error surface calculated with respect to the original data set, D^{obs}, then we obtain a revealing insight into the bootstrap distribution of the estimated parameters (see Figure 6.6). We can see that the bootstrap distribution provides quite a good reflection of the shape of the error surface. In particular, we can see that the small cluster centred around $a = 0.45, b = 0.8$ has a close correspondence to a local minimum in the error surface (in the a direction).

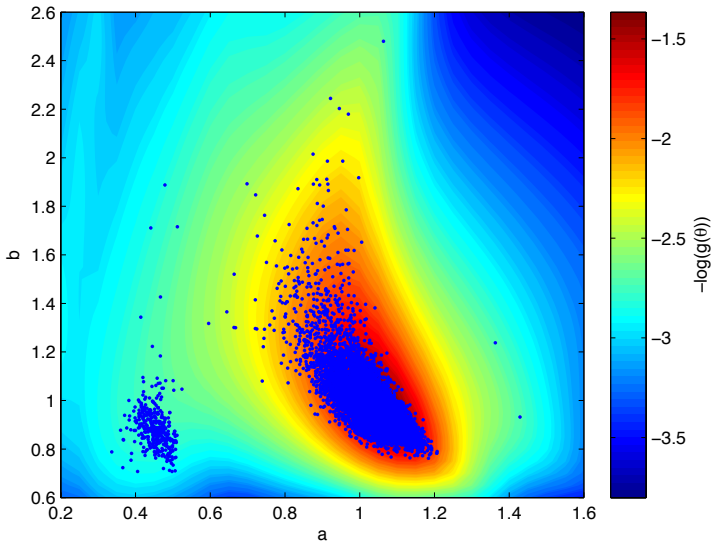


Figure 6.6: Joint distribution of bootstrap parameter estimates overlaid on top of the error surface of Figure 6.2. Note that we here colour the points of the scatterplot uniformly, to avoid confusion with the colour scale of the error surface contours.

6.3 Example II: JAK2-STAT5 signalling pathway

The JAK-STAT pathway is a well-studied signalling pathway that describes a mechanism by which signals carried by cytokines may be transduced to the cell nucleus via STAT activation, dimerisation, and relocation (Horvath, 2000; Aaronson and Horvath, 2002). In the case of the JAK2-STAT5 pathway, Epo (erythropoietin) triggers the activation of JAK2 kinases when it binds to EpoR (the Epo receptor), which then results in STAT5 becoming activated, dimerising, and moving to the cell nucleus. This process is illustrated in Figure 6.7.

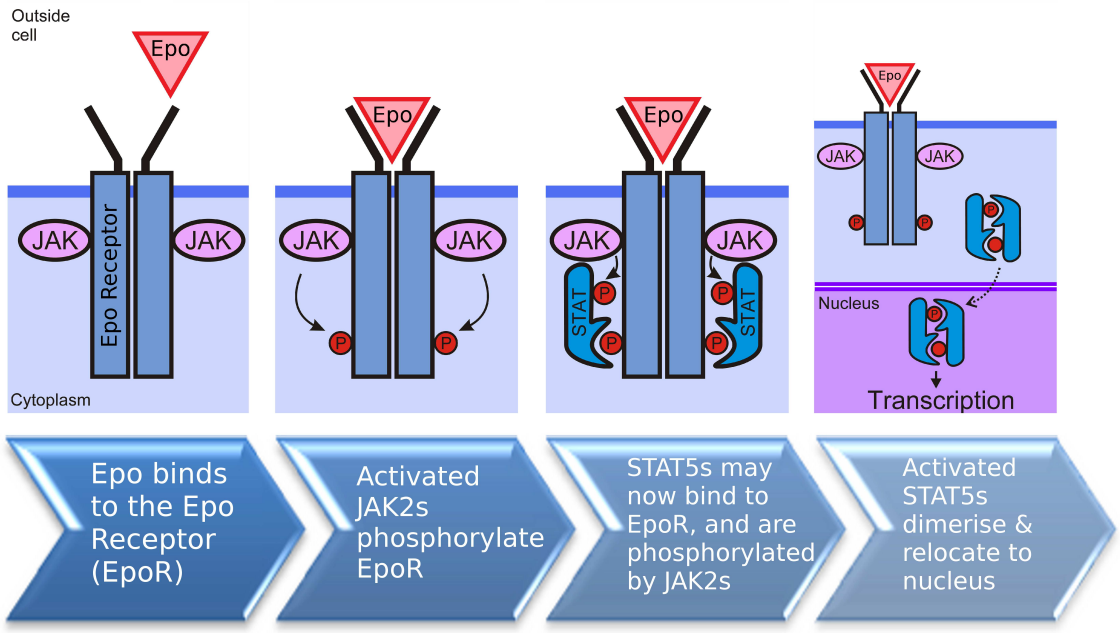


Figure 6.7: The JAK2-STAT5 signalling pathway (figure adapted from Znamenkiy, 2006).

Swameye *et al.* (2003) suggested a number of parametric ODE models to describe this signalling pathway, the parameters of which were estimated from experimental data. We consider one of the proposed models (taken from Swameye *et al.*, 2003, Supplementary Material), and – using data from the original experiments – apply the approach of Chapter 4 in order to obtain bootstrap distributions of the parameters.

6.3.1 The Model

The model we consider is as follows,

$$\begin{aligned} \frac{dv_1}{dt} &= -r_1 v_1 D + 2r_4 v_4 & \frac{dv_2}{dt} &= r_1 v_1 D - v_2^2 \\ \frac{dv_3}{dt} &= -r_3 v_3 + 0.5 v_2^2 & \frac{dv_4}{dt} &= r_3 v_3 - r_4 v_4. \end{aligned} \tag{6.4}$$

Here, v_1, v_2 and v_3 represent the concentrations in the cytoplasm of (respectively) unphosphorylated STAT5, phosphorylated monomeric STAT5, and phosphorylated dimeric STAT5. The variable v_4 denotes the concentration of STAT5 in the nucleus, and D is an experimentally determined quantity (which varies over time) related to the amount of Epo-induced phosphorylation of the EpoR (see Swameye *et al.*, 2003). The r_i's are parameters which are combinations of the rate constants of the system (see Swameye *et al.*, 2003, Supplementary Material). The initial values of v_2, v_3 and v_4 at time $t = 0$ are assumed to be zero (since it is supposed that all STAT5 in the cell is initially cytoplasmic and unphosphorylated), while the initial concentration of unphosphorylated cytoplasmic STAT5, $v_1(t = 0)$, is treated as an unknown parameter.

The quantities v_1, v_2, v_3 and v_4 could not be measured experimentally. Instead, the amount of phosphorylated STAT5 in the cytoplasm, y_1, and the total amount of cytoplasmic STAT5 (phosphorylated and unphosphorylated), y_2, were recorded. These can be written in terms of the v_i's as follows,

$$y_1 = r_5(v_2 + 2v_3) \tag{6.5}$$
$$y_2 = r_6(v_1 + v_2 + 2v_3), \tag{6.6}$$

where r_5 and r_6 are two unknown scaling parameters, which must also be estimated. In total, there are thus 6 unknown parameters in this model (r_1, r_3, r_4, r_5, r_6 and $v_1(0)$).

6.3.2 The data

Swameye *et al.* (2003) measured y_1 and y_2 at a number of discrete time points in order to obtain several sets of experimental data. We focus on just one of these (the "DATA1_hall" set, available from the original authors). The data are shown later in Figure 6.8.

6.3.3 Estimating the parameters

Given our concern that the optimisation routine employed in Section 6.2 may have failed (in a small number of cases) to have found the global optimum, we here employ a more sophisticated approach. We estimate the unknown parameters of the ODE system presented in (6.4) using the Stochastic Ranking Evolutionary Strategy (SRES) of Runarsson and Yao (2000), as implemented in the libSRES C library (Ji and Xu, 2006). This is a "global" optimisation routine that is (in principle) able to escape local optima (although, given a finite running time, this might not be the case in practice), and was recently found to provide the best performance in a test problem that sought to estimate 36 parameters of a nonlinear biochemical ODE model (Moles *et al.*, 2003). In order to improve our chances of locating the global optimum, we rerun the algorithm 8 times for each data set (and take as our final estimate the "best" amongst these 8 runs). We again seek the least squares estimate of the parameters, so employ the error function given in Equation (6.3).

SRES allows us to specify acceptable ranges for the parameter values, which we set to be: $v_1(0) \in [0.01, 10], r_1 \in [0.1, 10], r_3 \in [0.01, 5], r_4 \in [0.01, 5], r_5 \in [0.1, 10], r_6 \in [0.01, 10]$.

Before applying our bootstrap procedure, we use SRES to obtain parameter estimates from the original data set. These are (to 3 significant figures): $\hat{v}_1(0)^{\text{orig}} = 0.996$, $\hat{r}_1^{\text{orig}} = 2.43$, $\hat{r}_3^{\text{orig}} = 0.256$, $\hat{r}_4^{\text{orig}} = 0.303$, $\hat{r}_5^{\text{orig}} = 1.27$, $\hat{r}_6^{\text{orig}} = 0.944$.

6.3.4 Bootstrap procedure

As previously, we fit Gaussian process regression models independently to the y_1 data and to the y_2 data. We again employ a zero mean function and a squared exponential covariance function, and estimate the hyperparameters by maximisation of the marginal likelihood. The resulting fits are shown in Figure 6.8. Due to the computational costs associated with fitting using SRES, this time we sample only 1,500 replicate data sets from the model, and calculate parameter estimates for each one.

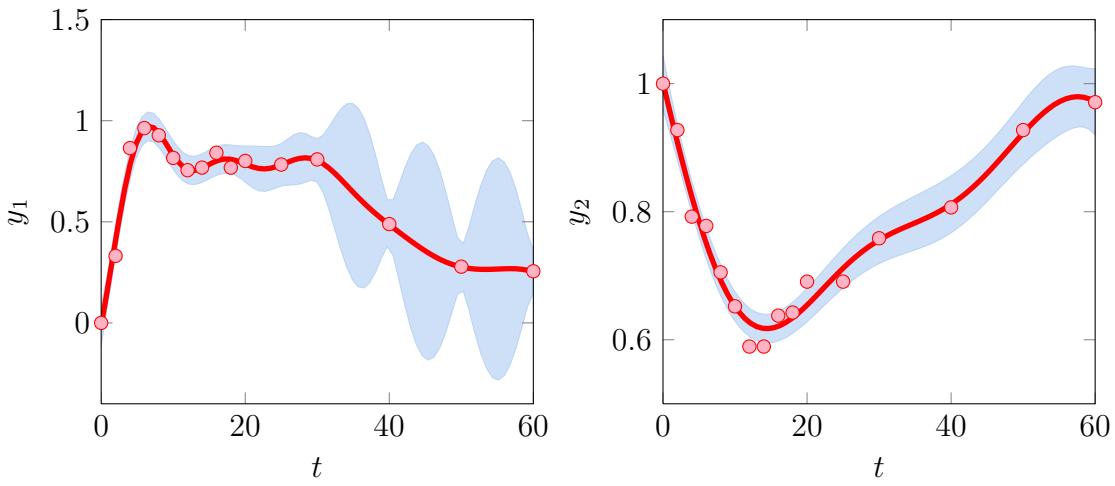


Figure 6.8: Gaussian process regression models fitted to the JAK2-STAT5 data, for y_1 (left) and y_2 (right). Data points are shown as filled red circles.

6.3.5 Results

Figure 6.9 shows that the marginal bootstrap distributions are generally centred around the original parameter estimates. However, for the r_3 parameter, there are two distinct clusters. The first (much larger) cluster comprises estimates centred around $\hat{r}_3^{\text{orig}}$. The second cluster comprises 28 estimates for which $r_3 \approx 5$.

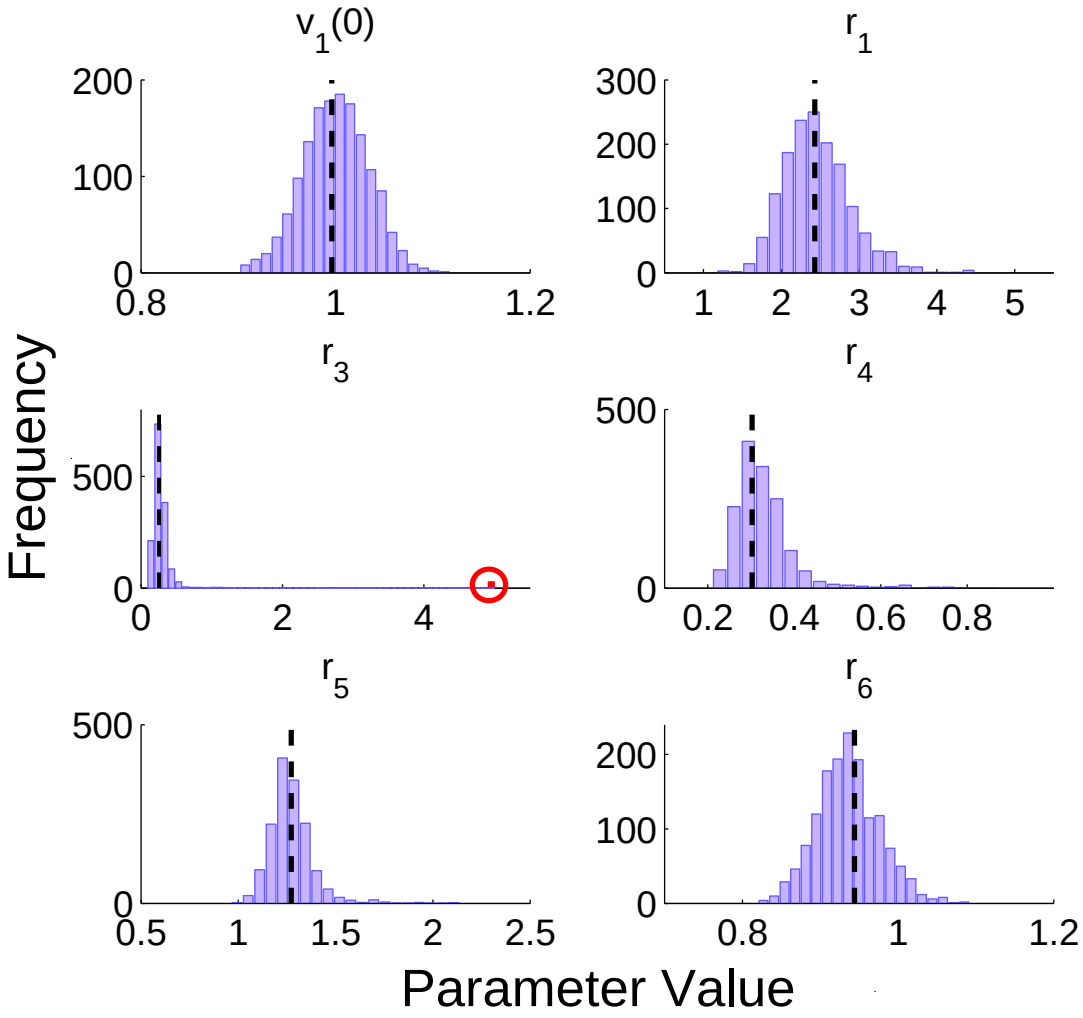


Figure 6.9: Histograms showing the marginal bootstrap distributions of the parameter estimates. The vertical black dashed lines indicate the parameter estimates obtained from the original data set. Although the distributions are generally quite narrow, note that for r_3 there is a small amount of probability mass located at $r_3 \approx 5$ (shown as a red bar and ringed by a red circle).

In Figure 6.10, we plot pairs of corresponding parameter estimates against one another. We show the 28 parameter estimates for which $r_3 \approx 5$ in red. It is not just the r_3 value that is unusual for these red points: we can see that the r_4 and r_5 values are also "extreme" (although less dramatically so than for r_3) .

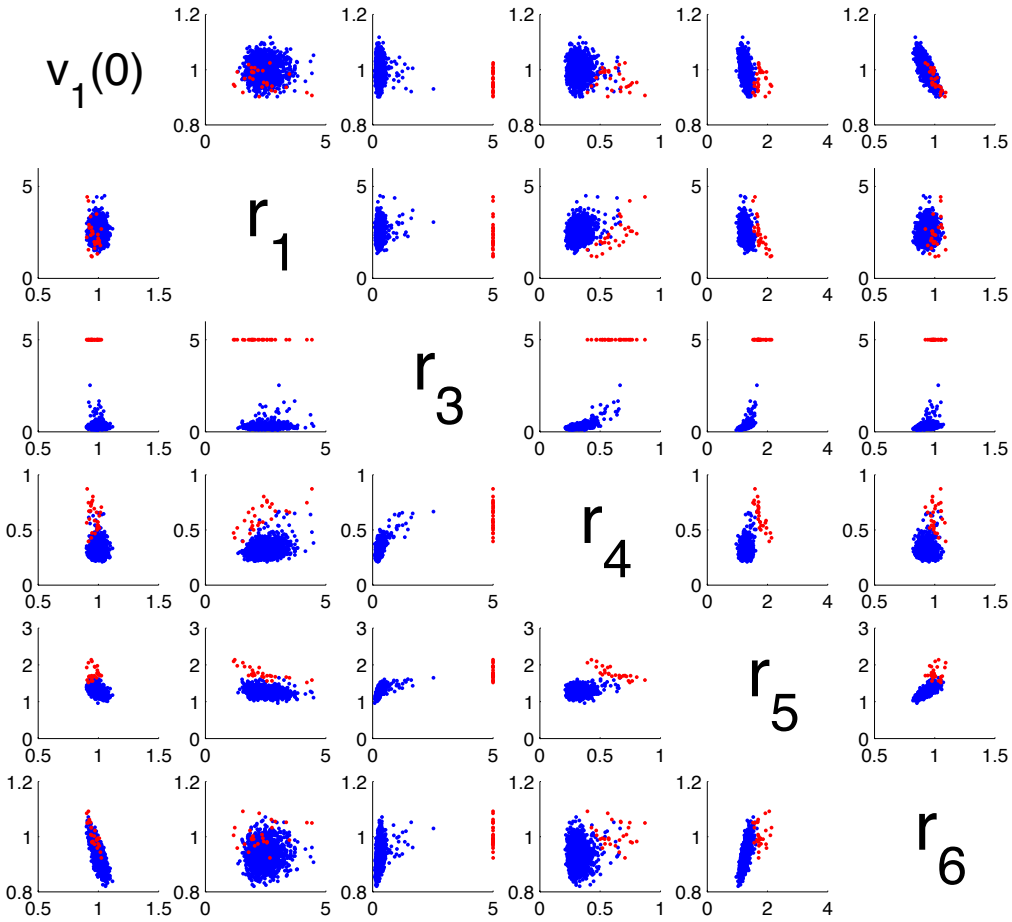


Figure 6.10: Scatterplots showing pairs of estimated parameter values. The y-axis of every plot in the top row corresponds to $v_1(0)$; the y-axis of every plot in the second row corresponds to r_1; ...and so on (as indicated by the labels). Similarly the x-axis of plots in the first column corresponds to $v_1(0)$, and so on. Red points correspond to parameter estimates in the "second set" (for which $r_3 \approx 5$).

It is reasonable to ask if the parameter estimates for which $r_3 \approx 5$ (the red points in Figure 6.10) arise as a result of a failure by the optimisation routine. However, this appears not to be the case, as the corresponding fits are comparable to the fits provided by the original estimates (see Figure 6.11). It follows that there is a second set of parameter values that are very different to the original estimates, but which nevertheless allow the ODE model to capture the observed behaviour.

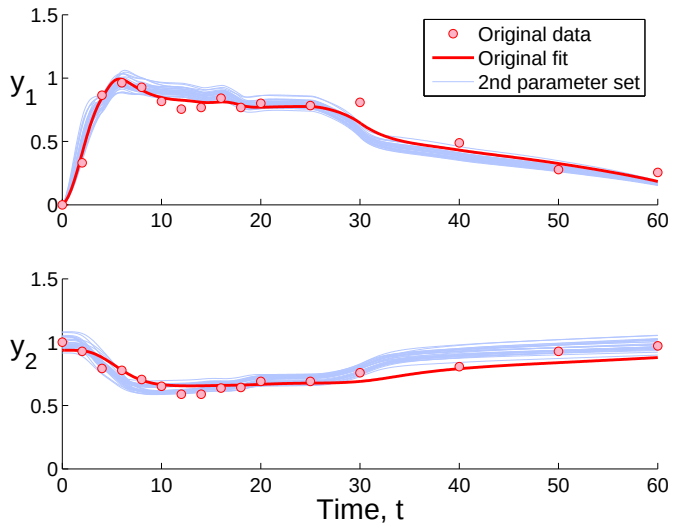


Figure 6.11: Plots showing the original experimental data set, original fit to this data set, and the fits obtained using values from the second set of parameter estimates.

6.4 A two-step approach

We mentioned in Section 6.1.2.2 that an alternative way in which to estimate the parameters of an ODE system is to fit a (typically nonparametric) model to the data, $\hat{\mathbf{x}}(t)$, and then to estimate the parameters $\boldsymbol{\theta}$ in order to minimise the discrepancy between $\frac{d}{dt}\hat{\mathbf{x}}(t)$ and $f(\mathbf{x}; \boldsymbol{\theta})$ (cf. Equation 6.1). As part of our bootstrap approach, we fit a GP regression model to the data. We could, therefore, make use of this as part of a two-step method for parameter estimation. We briefly illustrate this idea in the context of the Lotka-Volterra example of Section 6.2.

6.4.1 Sampling derivatives

One useful property of Gaussian processes is that the derivative of a GP is again a GP (see, for example Rasmussen and Williams, 2005). Recall that, given covariance function k for a GP, the covariance $\mathrm{cov}(f(t_i), f(t_j))$ is given by $k(t_i, t_j)$. The covariances between

the derivatives of f, and between function values and derivatives are then given by,

$$\operatorname{cov}\left(\left.\frac{df}{dt}\right|_{t=t_i}, \left.\frac{df}{dt}\right|_{t=t_j}\right) = \frac{d^2}{dt_i dt_j}k(t_i, t_j), \qquad \operatorname{cov}\left(f(t_i), \left.\frac{df}{dt}\right|_{t=t_j}\right) = \frac{d}{dt_j}k(t_i, t_j). \tag{6.7}$$

These results (and more general versions) are provided in Solak *et al.* (2003); Girard (2004); Rasmussen and Williams (2005). For brevity, we shall henceforth write $\frac{df}{dt_i}$ as shorthand for $\left.\frac{df}{dt}\right|_{t=t_i}$ and f_i as shorthand for $f(t_i)$. We consider below the specific case in which we are interested only in the function and derivative values at the times where we have observations. However, with minor modifications, our exposition may be extended to the more general case where we are interested in any finite collection of times.

If we assume a GP prior for f, with zero mean function and covariance function k, then,

$$\left[y_1, \ldots, y_r, f_1, \ldots, f_r, \frac{df}{dt_1}, \ldots, \frac{df}{dt_r}\right]^\top \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} K_o + \sigma_\epsilon^2 I & K_o & L_{FD} \\ K_o & K_o & L_{FD} \\ L_{DF} & L_{DF} & M \end{pmatrix}\right), \tag{6.8}$$

where y_i is the (noisy) observation obtained at time t_i and,

$$(K_o)_{ij} = k(t_i, t_j), \tag{6.9}$$

$$(L_{DF})_{ij} = \operatorname{cov}\left(\frac{df}{dt_i}, f_j\right), \tag{6.10}$$

$$(L_{FD})_{ij} = \operatorname{cov}\left(f_i, \frac{df}{dt_j}\right), \tag{6.11}$$

$$(M)_{ij} = \operatorname{cov}\left(\frac{df}{dt_i}, \frac{df}{dt_j}\right). \tag{6.12}$$

If we define,

$$K_{*o} = \begin{pmatrix} K_o \\ L_{DF} \end{pmatrix}, K_{o*} = \begin{pmatrix} K_o & L_{FD} \end{pmatrix}, \text{ and } K_{**} = \begin{pmatrix} K_o & L_{FD} \\ L_{DF} & M \end{pmatrix},$$

then we can see that Equation (6.8) bears a strong similarity to Equation (4.32). Using these definitions of K_{*o}, K_{o*} and K_{**}, it follows that,

$$\left[f_1, \ldots, f_r, \frac{df}{dt_1}, \ldots, \frac{df}{dt_r}\right]^\top \Bigg| [y_1, \ldots, y_r] \quad \sim \mathcal{N}(\mathbf{m}_{cond}, K_{cond}), \tag{6.13}$$

where $\mathbf{m}_{cond}$ and K_{cond} are as defined in Equations (4.34) and (4.35).

We may hence sample function values *and* corresponding derivative values from Equation (6.13).

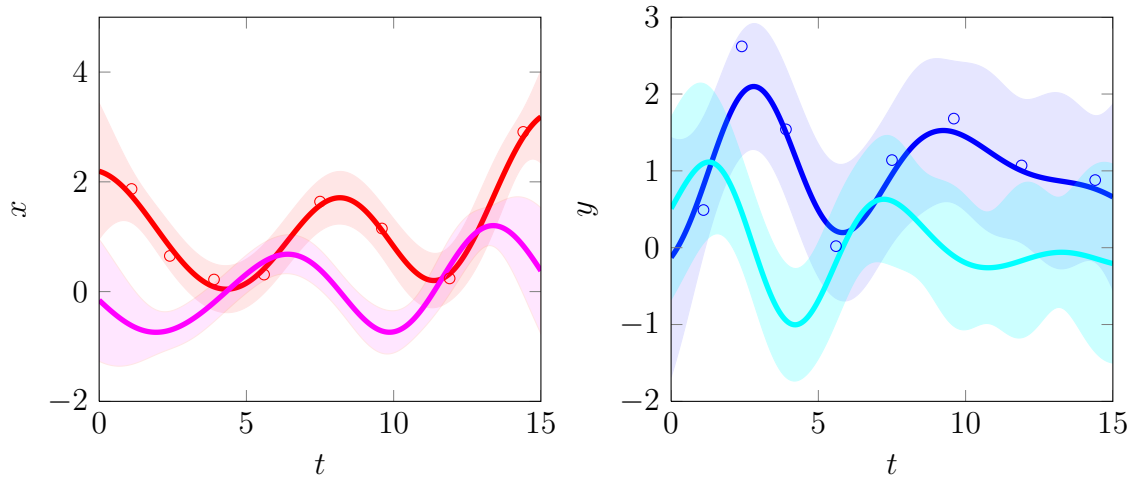


Figure 6.12: Left: Fitted GP model (red) and corresponding derivative process (magenta) for the x data. Right: Fitted GP model (blue) and derivative process (cyan) for the y data.

6.4.2 Sampling derivatives for the Lotka-Volterra example

We fit a GP regression model to the Lotka-Volterra data exactly as previously, but this time we additionally consider the derivative process, illustrated in Figure 6.12. We may then sample from the multivariate normal distribution of Equation (6.13) in order to obtain replicate function and corresponding derivative values. Figure 6.13 shows function and derivative values drawn from the multivariate Gaussian posterior, and compares to numerically estimated derivatives (note that, to aid visualisation, this figure actually illustrates the more general case where we permit sampling at time points other than those at which the observations were made). There is clearly good agreement between the sampled and numerically estimated derivatives.

6.4.3 Two-step estimation of Lotka-Volterra parameters

The procedure described in Sections 6.4.1 and 6.4.2, provides a way of sampling different approximations, $\hat{x}(t)$, to $x(t)$, and also their derivatives, $\frac{d}{dt}\hat{x}(t)$, all evaluated at the time points at which we have observations. Let us denote the i^{th} sampled approximation by $\hat{x}_i(t)$, and also define $\hat{\mathbf{x}}_i := [\hat{x}_i(t_1), \ldots, \hat{x}_i(t_n)]^\top$ to be the vector comprising the values of $\hat{x}_i(t)$ evaluated at the time points $t_1, \ldots, t_n$ at which we have observations. We define $\hat{\mathbf{y}}_i$ similarly.

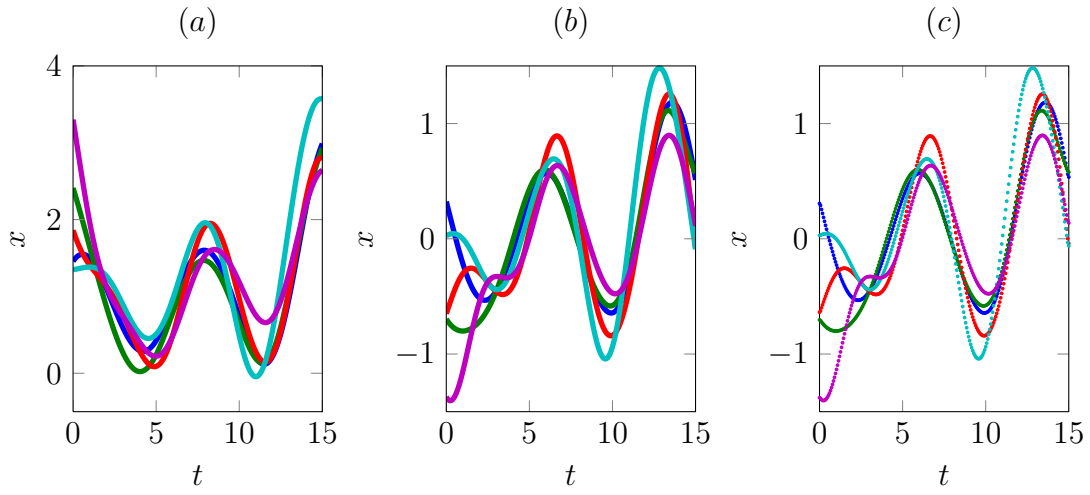


Figure 6.13: (a) Functions and (b) corresponding derivatives sampled from the joint distribution of function and derivative values for the x data (cf. Figure 6.12, left). (c) Derivatives estimated numerically from the curves in (a) using the `gradient` function in Matlab (MATLAB, 2009). Note that the colours match up between (a), (b) and (c), so that — for example — the red curves in (b) and (c) represent the derivative of the red curve in (a).

Note that we may rearrange Equation (6.2) in order to obtain,

$$xy + \frac{dx}{dt} = ax \tag{6.14}$$

$$y - \frac{dy}{dt} = bxy. \tag{6.15}$$

Since we have approximations to all terms in Equations (6.14) and (6.15) *except* a and b, and since also both equations are linear in the parameters, we may now find least squares estimates of a and b analytically. For example, if we plug our approximations into Equation (6.14) and rewrite as $\hat{\mathbf{z}}_1 = a\hat{\mathbf{z}}_2$ (where $\hat{\mathbf{z}}_2$ is our approximation to x, and $\hat{\mathbf{z}}_1$ is our approximation to the lefthand side of Equation 6.14), then the least squares estimate for a is given by $\hat{a} = \hat{\mathbf{z}}_1/\hat{\mathbf{z}}_2$. We may estimate b similarly. It follows that for each collection of approximations, $\{\hat{\mathbf{x}}_i, \frac{d}{dt}\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \frac{d}{dt}\hat{\mathbf{y}}_i\}$, we may obtain estimates of a and b analytically (and hence very quickly).

We sample 10,000 pairs, $(\hat{\mathbf{x}}_i, \frac{d}{dt}\hat{\mathbf{x}}_i)$, of approximations for $x(t)$ and $\frac{d}{dt}\hat{\mathbf{x}}_i$, and 10,000 pairs, $(\hat{\mathbf{y}}_i, \frac{d}{dt}\hat{\mathbf{y}}_i)$, for $y(t)$ and $\frac{d}{dt}\hat{\mathbf{y}}_i$. For each collection, $\{\hat{\mathbf{x}}_i, \frac{d}{dt}\hat{\mathbf{x}}_i, \hat{\mathbf{y}}_i, \frac{d}{dt}\hat{\mathbf{y}}_i\}$, we estimate the parameters as described above. The resulting marginal distributions are shown in Figure 6.14.

The marginal distributions produced using the two-step approach are clearly very similar to those shown in Figure 6.5, but are obtained at a vastly reduced computational cost. The total length of time required to obtain the bootstrap distributions of Figure 6.5 was 5696 seconds (approximately 1 hour 35 minutes). In contrast, it took just 1.5 seconds to generate the marginal distributions shown in Figure 6.14 (and, in both cases, the quoted times include fitting and sampling from the Gaussian process regression models).

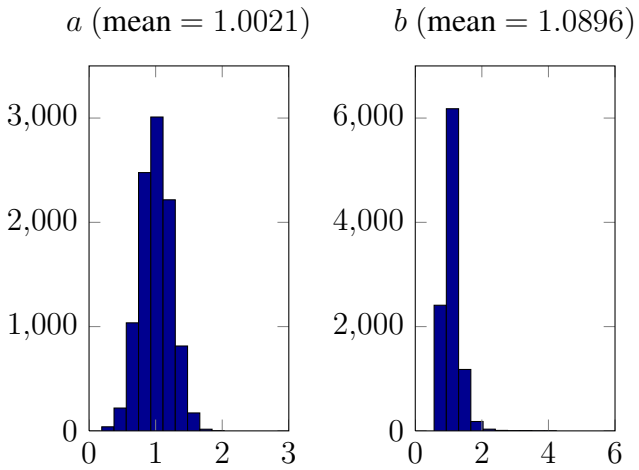


Figure 6.14: Histograms of the marginal distributions of the estimates for a (left) and b (right), obtained for the Lotka-Volterra example using the two-step procedure of Section 6.4.

6.5 Discussion

We applied the bootstrap procedure of Chapter 4 in the context of parameter estimation for ODE models. We first demonstrated the utility of this approach by considering a simulated Lotka-Volterra model. By bootstrapping the simulated time course data, we were able to obtain distributions of parameter estimates, which reflect the effects of variability in the data. The majority of the estimates were clustered around $\hat{a}^{\text{orig}}$ and $\hat{b}^{\text{orig}}$ (the estimates from the original data set). However, there was also a distinct second cluster. The estimates in this second cluster generally provided worse fits to the data than those in the first, which suggested a possible failure of the optimisation algorithm. In spite of this, we note that the mean values of the bootstrap distributions were closer than the original estimates to the values that were used to simulate the data ($a = 1, b = 1$).

We then considered a model of the JAK2-STAT5 signalling pathway, which we fitted to real experimental data. We obtained bootstrap distributions of the parameter estimates, and again observed two clusters of parameter estimates. This time, however, the smaller cluster appeared to be genuine, as the parameter estimates belonging to this cluster provided a very good fit to the original data set (comparable to the fits provided by the estimates in the larger cluster).

We finally considered how the Gaussian process regression model used to perform the bootstrap might also be used as part of the estimation procedure itself, and demonstrated that this approach provided significant computational gains, with apparently no loss in the quality of estimation. Indeed, the mean values of the marginal distributions shown in Figure 6.14 were very close to the parameter values that were used to generate the data. Since we were able to obtain parameter estimates analytically, this also avoided the problems with local minima that we had seen previously. However, it is important to note that we were only able to obtain analytical solutions because the ODE model was linear

in the parameters. In general, this will not be the case, and hence for most practical applications we would be forced to use numerical optimisation. Nevertheless, optimisation in the "two-step" case may reasonably be expected to be faster than optimisation in the "simulation" case, since we avoid having to solve the ODE. It should also be noted that we do not need to know the initial conditions of the system in order to apply two-step approaches, which is in stark contrast to simulation methods. The biggest limitation of the approach as currently presented is that we require observations of *all* state variables, which will rarely be the case for more realistic examples. Extending the methodology to cope with incompletely observed systems could be an interesting direction for future work.

Perhaps the most important message of this chapter is that single point estimates of the parameters are of limited value, providing no information whatsoever about stability to data perturbations. Given that biological data *are* noisy, this would seem to be woefully inadequate. Although we maintain that bootstrap approaches are useful for addressing this problem (and allow us to extend the utility of existing point-estimate methods), Bayesian methods (in which we place priors directly over the unknown parameters) are also applicable. Since, in the bootstrap case, we will usually require global optimisation routines in order to estimate the parameters, the usual criticism of Bayesian approaches being too computationally costly would be unfair. We therefore present our bootstrap procedure as a useful alternative to a full Bayesian treatment.

We note that, contemporaneous with our investigation into the two-step approach of Section 6.4, a similar method was proposed by Calderhead *et al.* (2009) from a Bayesian perspective. These authors provide a rather more sophisticated procedure than the one presented here, combining the potentially conflicting derivative approximations $\frac{d}{dt}\hat{\mathbf{x}}$ and $h(\hat{\mathbf{x}}; \boldsymbol{\theta})$ as a *product of experts* (Mayraz and Hinton, 2002). Samples from the posterior parameter distribution are obtained using a Markov chain Monte Carlo method (Jasra *et al.*, 2007). Again, we regard our approach as an alternative, which — while less sophisticated — may be used as a wrapper around existing methods for parameter estimation, and hence might (in some circumstances) be easier to implement.

Chapter 7

Discussion

7.1 Summary

Throughout this thesis, we have been concerned with methods for approximating data generating processes (DGPs), and how the resulting approximations may be used in order to assess the stability with which quantities and structures are inferred from systems biology data.

In Chapters 2 and 3 we focused upon covariate selection in the context of biomarker discovery problems, and considered how subsampling may be used to quantify selection stability. We thereby proposed a novel algorithm for determining a final set of stably selected covariates, and used a simulation study to demonstrate that our approach has favourable properties in terms of the number of "false positive" selections that are made (Chapter 2). We showed that correlations amongst covariates may cause difficulties for stability selection approaches that employ the lasso (Section 2.4.2). However, by using our approach in tandem with the elastic net likelihood penalty, we were able to mitigate these challenges. In Chapter 3, we applied our approach to a problem in HTLV-1 proteomic biomarker discovery, and identified a number of putative SELDI peak biomarkers for the inflammatory condition HAM/TSP. Two of these have been experimentally identified as Calgranulin B and β_2-microglobulin, both of which seem highly plausible biomarkers.

In Chapter 4 we proposed a novel method for bootstrapping time courses of data. Our approach is in some ways similar to existing methods for bootstrapping residuals, but additionally employs concepts from Gaussian process regression (GPR) in order to capture the uncertainty in the unknown regression model. We derived our procedure from conjugate Bayesian and Gaussian process regression standpoints, and demonstrated that both derivations lead to the same approximation to the DGP.

In Chapter 5 we applied the multivariate Gaussian bootstrap of Chapter 4 in order to assess the stability of networks inferred from gene expression time courses using the "GeneNet"

algorithm of Schäfer *et al.* (2006). We suggested different methods for bootstrapping depending upon whether the time course data were viewed as cross-sectional or longitudinal. We applied the procedure of Chapter 4 and also nonparametric bootstrap approaches to an *Arabidopsis thaliana* gene expression data set. Regardless of the method used to bootstrap the data, the networks we obtained using GeneNet appeared relatively unstable (as quantified by using a similarity measure to assess the concordance between the replicate and original networks). Nevertheless, there were some edges in the network that had high levels of bootstrap support. We therefore suggested that bootstrap approaches such as the ones we considered might be use to generate stable subnets, which are present in all (or most) of the replicate networks. We finally discussed the possible effects of having many correlated genes, and the impact that these might have upon stability.

In Chapter 6, we applied our multivariate Gaussian bootstrap in the context of ordinary differential equation (ODE) parameter estimation. For illustration, we briefly considered a simulation example based upon a Lotka-Volterra model, and demonstrated the utility of our approach. We then moved on to a model of the JAK2-STAT5 cell signalling pathway. We found that parameter estimates were generally fairly stable, with the majority of our replicate estimates being close to those obtained from the original data set. However, we also identified a distinct second set of parameters that provided a good fit to the original data, but were very different to the original estimates. Finally, we exploited the Gaussian process regression derivation of our bootstrap approach, and incorporated the underlying GPR model within the estimation procedure. Returning to our earlier Lotka-Volterra example, we obtained very similar results, but at a massively reduced computational cost.

7.2 Conclusions

Assessing the stability of inferences to data perturbations is clearly very important, particularly in light of the many sources of variability present in current biological studies (see Chapter 1). As exemplified by reanalyses of early SELDI-TOF-MS studies (Baggerly *et al.*, 2004), conclusions drawn from biomedical data can be difficult or impossible to reproduce, which raises serious questions about their validity. The approaches we have presented in this thesis are designed either to identify the most stable conclusions, or at least to quantify the stability of our inferences. Bootstrapping, subsampling, and similar procedures for approximating the underlying DGP have a relatively long history in biology (e.g. Felsenstein, 1985). Although Bayesian approaches are increasingly popular and are incredibly useful for quantifying uncertainty, we nevertheless believe that bootstrapping remains an important, effective and practical tool, and take the view that it is useful to have a variety of techniques available to us. The most obvious benefit of bootstrapping is a practical one: the "plug-in principle" of Efron and Tibshirani (1993) means that any existing procedure for drawing conclusions from our observed data may also be applied to our replicate data sets. This is particularly useful given the availability of computer programs that take a data file as input and then return an output (without us necessarily knowing the details of the algorithms employed).

In Chapter 2, we clearly demonstrated the difficulties (in terms of stability) caused by feature selection algorithms such as the lasso, which only pick out a single representative from a correlated set of covariates. One of the main contributions of this chapter was therefore to provide a procedure that would allow different feature selection algorithms to be considered together, which allowed us to explore different tolerances to the selection of several correlated covariates by using the elastic net. A further novelty described in Chapter 2 was the use of a probabilistic score that combined assessments of stability and predictive performance. For classification problems, this represents an alternative to the procedure described by Meinshausen and Bühlmann (2010), which instead seeks to control the proportion of "falsely selected" covariates. Since the distinction between "relevant" and "noise" (i.e. "irrelevant") covariates might not be clear-cut in practical applications, we believe that incorporating an assessment of predictive performance might provide a more pragmatic approach. However, our approach is limited to classification problems, and it is unclear how it might be extended to the regression case. The main challenge is to devise a principled means to combine an assessment of predictive performance for a regression model with an assessment of stability. However, in the context of biomarker discovery (our main focus), this limitation is a relatively minor one, since we are usually presented with case/control data rather than continuous outcomes.

In Chapter 3, we applied our approach to the problem of identifying SELDI-TOF-MS peaks whose intensities allowed us to distinguish between sufferers of HAM/TSP and asymptomatic carriers (ACs). Our current understanding of the pathogenesis of HAM/TSP is incomplete, and hence this novel proteomic data set provided an opportunity to shed further light on the development of disease. Our analysis identified several putative biomarkers, two of which have been experimentally identified. These two proteins are well-known markers of inflammatory conditions (Xie and Yi, 2003; Foell and Roth, 2004), which has both negative and positive implications. On the one hand, it means that we have not (yet) identified any protein biomarkers that are specific to HAM/TSP, rather than being general markers of inflammation (although we note that the *abundances* of these proteins might vary significantly between HAM/TSP and other inflammatory conditions; this remains to be established). On the other hand, it implies that our results are highly plausible, and adds further weight to the argument that additional experimental resources should be allocated to identify the other putative biomarkers. At the very least, we have added HAM/TSP to the list of inflammatory conditions with which these proteins are associated.

One additional outcome of our work with SELDI-TOF-MS data was to establish that these data *can* provide useful results. As discussed in Chapter 3, as a result of the work of Baggerly *et al.* (2004), there has been a degree of concern about the viability of conclusions drawn from SELDI-TOF-MS data. However, the difficulties identified with previous analyses concerned experimental design and statistical methodologies. Provided appropriate precautions are taken (randomising samples across chips, using consistent and appropriate normalisation procedures, and — ideally — treating the SELDI-TOF-MS analysis as a filtering step in a larger protein identification study), these difficulties may be avoided.

The work of Chapter 4 and subsequent chapters focused upon the use of a multivariate

Gaussian model to perform a bootstrap of time course data. Of paramount importance for these approaches was the use of a function, k, to model the covariance structure of the data. When applied to gene expression time course data (Chapter 5), the relative instability of networks inferred using GeneNet was striking. We posited that this instability might be due to the effects of correlation amongst the gene expression time courses. However, this remains a hypothesis, and it would be useful to perform comprehensive simulation studies in order to confirm or reject this possibility. Such a method could also be applied in order to determine the effects of small sample sizes upon network stability. In the *Arabidopsis thaliana* example that we considered, there were only two time courses for each gene. Having a larger number of replicates might allow us to estimate the variability in these time courses (and resulting partial correlation estimates) more accurately. We believe that the nonparametric bootstrap approaches that we considered are liable to *under*estimate the variability in the data (simply because, as a result of there being so few replicates, there are relatively few different bootstrap data sets for each time course). Thus, the stability results that we obtained using these methods might actually be overoptimistic.

Applying our multivariate Gaussian bootstrap in the context of ODE parameter estimation (Chapter 6) yielded some interesting results. The existence of a second set of plausible parameter estimates for the JAK2-STAT5 model provides an excellent illustration of the importance of quantifying the uncertainty in our inferences. Bayesian approaches (in which we specify and then update a prior for the parameters) would also be applicable here (Toni *et al.*, 2009; Calderhead *et al.*, 2009). Our discovery of a second plausible parameter set for this model might manifest itself as a bimodal likelihood surface and hence, if a flat prior is assumed, a bimodal posterior. The use of Gaussian process regression models in order to speed up parameter estimation for ODE systems (Section 6.4) would seem to be a worthwhile pursuit, and we are pleased that the approach of Calderhead *et al.* (2009) has been demonstrated to provide significant computational savings relative to alternative methods.

7.3 Further work

A number of possible directions for further work have been suggested throughout this thesis. In the immediate future, we will extend the results of Chapter 3 by considering recently generated data comparing patients with HAM/TSP to those with multiple sclerosis (MS). Additionally, experimental work is ongoing in order to identify more of the protein peaks selected in Chapter 3.

The apparent instability of networks inferred from gene expression data (Chapter 5) demands further investigation. As we have suggested, devising and conducting a controlled simulation study to probe the properties of this and similar inference algorithms would seem a useful means to establish their strengths and limitations. This is in good agreement with the sentiments of (Werhli *et al.*, 2006), who also advocate the use of simulation

studies to assess the utility of network inference algorithms.

The multivariate Gaussian bootstrap approach of Chapter 3 could be extended in a number of obvious ways. Firstly, the current approach is limited to treating time course data sets independently of one another. Procedures exist for fitting dependent Gaussian process regression models (Boyle and Frean, 2005), and we could apply these when constructing our bootstrap model. Secondly, in Section 4.2.4, we only considered hyperparameter estimation by maximisation of the marginal likelihood. An alternative approach would be to infer these parameters using Markov chain Monte Carlo sampling (as in Neal, 1999). This would add to the computational cost of our approach, and it is unclear whether or not it would make a significant difference to the results. Nevertheless, it might be an interesting alternative to explore.

Throughout this thesis, we have been concerned with the approximation of DGPs, either by using subsampling or bootstrap procedures. A related concept that has been recently proposed is *stochastic emulation* (Henderson *et al.*, 2009). Here, the "observed data" comprise samples generated by simulating from a stochastic model (using, for example, the stochastic simulation algorithm of Gillespie 1977). Crucially, we simulate these data using a variety of different values for the stochastic model's inputs (typically its parameters). The aim of emulation is to approximate the stochastic model, so that, given input $\mathbf{x}$, we can sample ("emulate") a corresponding output, $\mathbf{y}$. The idea is that, once we have such an approximation, we can dispense with the original stochastic model (from which it is typically much slower to generate samples). This problem has previously been considered in the context of deterministic models (Kennedy and O'Hagan, 2001; Conti and O'Hagan, 2010), but stochastic systems remain relatively unexplored (with the exception of Henderson *et al.* 2009). We believe that stochastic emulation represents an interesting future direction, which we believe may be of particular value given the current popularity of simulation-based inference procedures (Marjoram *et al.*, 2003; Sisson *et al.*, 2007). For this reason, we include in Appendix B a pilot study that explores stochastic emulation, which we hope will prove valuable for future work in this area.

7.4 Final remark

In addition to showing the benefits of including stability amongst the objectives that we seek to attain (Chapter 2), we have also demonstrated that conclusions drawn from current biological data can be unstable (as illustrated in Chapter 5 in the case of gene networks), and that small perturbations to the data can have significant effects upon the quantities derived from them (as we saw for the parameter estimates of the JAK2-STAT5 model in Chapter 6). Our final remark must therefore be to reiterate the importance of assessing the stability of conclusions drawn from experimental data (whether using the approaches considered here or any other techniques).

Appendix A

Node labels for Chapter 5

For completeness, we provide over the next few pages the sequence IDs corresponding to the node labels used in Chapter 5, Figure 5.4. Sequence annotations and further information may be found by visiting the Plant Expression Database (Wise *et al.*, 2007).

Node number (#)	Sequence ID (ID)
1	AFFX-Athal-GAPDH_3_s_at
2	AFFX-Athal-Actin_3_f_at
3	267612_at
4	267520_at
5	267517_at
6	267516_at
7	267456_at
8	267454_at
9	267432_at
10	267423_at
11	267383_at
12	267377_at
13	267341_at
14	267274_at
15	267262_at
16	267123_at
17	267063_at
18	267005_at
19	267000_at
20	266995_at
21	266993_at
22	266991_at
23	266928_at
24	266927_at
25	266925_at
26	266897_at
27	266864_at
28	266835_at
29	266820_at
30	266813_at
31	266809_at
32	266805_at
33	266789_at
34	266719_at
35	266671_at
36	266572_at
37	266511_at
38	266481_at
39	266460_at

#	ID	#	ID	#	ID	#	ID
40	266458_at	80	264930_at	120	263852_at	160	262940_at
41	266437_at	81	264924_at	121	263805_at	161	262888_at
42	266417_at	82	264916_at	122	263796_at	162	262885_at
43	266399_at	83	264901_at	123	263780_at	163	262882_at
44	266314_at	84	264838_at	124	263779_at	164	262879_at
45	266297_at	85	264837_at	125	263761_at	165	262875_at
46	266293_at	86	264832_at	126	263717_at	166	262877_at
47	266247_at	87	264820_at	127	263711_at	167	262826_at
48	266235_at	88	264806_at	128	263696_at	168	262786_at
49	266226_at	89	264779_at	129	263668_at	169	262784_at
50	266172_at	90	264774_at	130	263664_at	170	262748_at
51	266139_at	91	264728_at	131	263583_at	171	262717_s_at
52	266089_at	92	264738_at	132	263529_at	172	262644_at
53	266065_at	93	264585_at	133	263497_at	173	262635_at
54	266059_at	94	264580_at	134	263495_at	174	262626_at
55	265998_at	95	264553_s_at	135	263489_at	175	262609_at
56	265968_at	96	264479_at	136	263473_at	176	262604_at
57	265892_at	97	264428_at	137	263461_at	177	262597_at
58	265857_s_at	98	264427_at	138	263460_at	178	262569_at
59	265842_at	99	264383_at	139	263448_at	179	262504_at
60	265768_at	100	264382_at	140	263433_at	180	262503_at
61	265721_at	101	264355_at	141	263426_at	181	262501_at
62	265695_at	102	264313_at	142	263410_at	182	262498_at
63	265674_at	103	264262_at	143	263375_s_at	183	262473_at
64	265646_at	104	264261_at	144	263296_at	184	262455_at
65	265569_at	105	264250_at	145	263287_at	185	262432_at
66	265480_at	106	264211_at	146	263264_at	186	262426_s_at
67	265474_at	107	264207_at	147	263255_at	187	262407_at
68	265386_at	108	264179_at	148	263252_at	188	262343_at
69	265338_at	109	264131_at	149	263250_at	189	262341_at
70	265309_at	110	264102_at	150	263243_at	190	262295_at
71	265287_at	111	264063_at	151	263209_at	191	262277_at
72	265281_at	112	264061_at	152	263193_at	192	262232_at
73	265248_at	113	264057_at	153	263158_at	193	262201_at
74	265244_at	114	264038_at	154	263115_at	194	262194_at
75	265182_at	115	264004_at	155	263047_at	195	262174_at
76	265097_at	116	263985_at	156	263010_at	196	262173_at
77	265078_at	117	263906_at	157	262996_at	197	262164_at
78	264986_at	118	263882_at	158	262988_at	198	262134_at
79	264959_at	119	263880_at	159	262978_at	199	262127_at

#	ID	#	ID	#	ID	#	ID
200	262089_s_at	240	261353_at	280	260036_at	320	259111_at
201	261958_at	241	261350_at	281	260028_at	321	259081_at
202	261956_at	242	261255_at	282	260026_at	322	259070_at
203	261951_at	243	261252_at	283	260010_at	323	259068_at
204	261949_at	244	261206_at	284	259983_at	324	259069_at
205	261945_at	245	261167_at	285	259950_at	325	258972_at
206	261920_at	246	261129_at	286	259943_at	326	258958_at
207	261913_at	247	261122_at	287	259936_at	327	258925_at
208	261904_at	248	261080_at	288	259927_at	328	258871_at
209	261895_at	249	261059_at	289	259875_s_at	329	258781_at
210	261827_at	250	261046_at	290	259860_at	330	258771_at
211	261810_at	251	261032_at	291	259840_at	331	258764_at
212	261792_at	252	260913_at	292	259821_at	332	258749_at
213	261791_at	253	260896_at	293	259791_at	333	258736_at
214	261790_at	254	260837_at	294	259789_at	334	258729_at
215	261774_at	255	260831_at	295	259768_at	335	258724_at
216	261772_at	256	260794_at	296	259757_at	336	258723_at
217	261767_s_at	257	260727_at	297	259681_at	337	258622_at
218	261715_at	258	260725_at	298	259669_at	338	258614_at
219	261696_at	259	260693_at	299	259645_at	339	258497_at
220	261692_at	260	260676_at	300	259538_at	340	258463_at
221	261663_at	261	260602_at	301	259511_at	341	258432_at
222	261661_at	262	260590_at	302	259488_at	342	258379_at
223	261642_at	263	260570_at	303	259460_at	343	258350_at
224	261635_at	264	260569_at	304	259406_at	344	258315_at
225	261639_at	265	260566_at	305	259396_at	345	258249_s_at
226	261576_at	266	260552_at	306	259395_at	346	258244_at
227	261569_at	267	260455_at	307	259373_at	347	258218_at
228	261530_at	268	260412_at	308	259363_at	348	258196_at
229	261486_at	269	260390_at	309	259357_at	349	258188_at
230	261484_at	270	260380_at	310	259354_at	350	258181_at
231	261457_at	271	260308_at	311	259295_at	351	258150_at
232	261456_at	272	260143_at	312	259277_at	352	258089_at
233	261440_at	273	260137_at	313	259275_at	353	258060_at
234	261425_at	274	260125_at	314	259224_at	354	258054_at
235	261417_at	275	260116_at	315	259194_at	355	257985_at
236	261407_at	276	260099_at	316	259185_at	356	257984_at
237	261379_at	277	260075_at	317	259140_at	357	257933_at
238	261368_at	278	260055_at	318	259131_at	358	257911_at
239	261355_at	279	260045_at	319	259123_at	359	257849_at

#	ID	#	ID	#	ID	#	ID
360	257790_at	400	256541_at	440	255087_at	480	254162_at
361	257744_at	401	256527_at	441	255070_at	481	254125_at
362	257743_at	402	256524_at	442	255039_at	482	254083_at
363	257730_at	403	256480_at	443	255012_at	483	254053_s_at
364	257722_at	404	256468_at	444	254930_at	484	254034_at
365	257710_at	405	256456_at	445	254923_at	485	253951_at
366	257714_at	406	256441_at	446	254891_at	486	253949_at
367	257705_at	407	256322_at	447	254874_at	487	253946_at
368	257410_at	408	256266_at	448	254862_at	488	253928_at
369	257311_at	409	256263_at	449	254859_at	489	253927_at
370	257285_at	410	256233_at	450	254804_at	490	253926_at
371	257269_at	411	256232_at	451	254790_at	491	253922_at
372	257252_at	412	256216_at	452	254785_at	492	253876_at
373	257237_at	413	256198_at	453	254746_at	493	253838_at
374	257235_at	414	256169_at	454	254715_at	494	253776_at
375	257222_at	415	256100_at	455	254705_at	495	253730_at
376	257204_at	416	256096_at	456	254691_at	496	253708_at
377	257193_at	417	256076_at	457	254687_at	497	253702_at
378	257188_at	418	256057_at	458	254684_at	498	253700_at
379	257131_at	419	256049_at	459	254657_s_at	499	253695_at
380	257101_at	420	256020_at	460	254642_at	500	253693_at
381	257057_at	421	255982_at	461	254594_at	501	253636_at
382	257045_at	422	255844_at	462	254549_at	502	253592_at
383	257044_at	423	255827_at	463	254530_at	503	253581_at
384	257021_at	424	255764_at	464	254515_at	504	253577_at
385	256984_at	425	255763_at	465	254496_at	505	253559_at
386	256856_at	426	255723_at	466	254448_at	506	253550_at
387	256819_at	427	255716_at	467	254376_at	507	253548_at
388	256787_at	428	255674_at	468	254371_at	508	253523_at
389	256751_at	429	255645_at	469	254356_at	509	253484_at
390	256746_at	430	255614_at	470	254328_at	510	253460_at
391	256725_at	431	255572_at	471	254306_at	511	253438_at
392	256676_at	432	255513_at	472	254298_at	512	253425_at
393	256666_at	433	255479_at	473	254275_at	513	253394_at
394	256655_at	434	255457_at	474	254262_at	514	253331_at
395	256626_at	435	255455_at	475	254250_at	515	253252_at
396	256596_at	436	255437_at	476	254239_at	516	253251_at
397	256548_at	437	255325_at	477	254233_at	517	253243_at
398	256547_at	438	255304_at	478	254227_at	518	253235_at
399	256543_at	439	255104_at	479	254211_at	519	253202_at

#	ID	#	ID	#	ID	#	ID
520	253200_at	560	251786_at	600	250565_at	640	249508_at
521	253174_at	561	251775_s_at	601	250563_at	641	249470_at
522	253116_at	562	251768_at	602	250529_at	642	249411_at
523	253059_s_at	563	251758_at	603	250520_at	643	249385_at
524	253039_at	564	251753_at	604	250477_at	644	249377_at
525	252981_at	565	251742_at	605	250439_at	645	249355_at
526	252917_at	566	251730_at	606	250433_at	646	249354_at
527	252915_at	567	251673_at	607	250423_s_at	647	249327_at
528	252880_at	568	251664_at	608	250408_at	648	249315_at
529	252859_at	569	251658_at	609	250394_at	649	249276_at
530	252785_at	570	251598_at	610	250261_at	650	249230_at
531	252678_s_at	571	251524_at	611	250254_at	651	249211_at
532	252625_at	572	251519_at	612	250253_at	652	249134_at
533	252603_at	573	251483_at	613	250243_at	653	249122_at
534	252562_s_at	574	251391_at	614	250226_at	654	249046_at
535	252481_at	575	251338_at	615	250217_at	655	249002_at
536	252468_at	576	251326_at	616	250155_at	656	248984_at
537	252442_at	577	251322_at	617	250097_at	657	248953_at
538	252429_at	578	251324_at	618	250088_at	658	248952_at
539	252427_at	579	251310_at	619	250033_at	659	248910_at
540	252420_at	580	251287_at	620	250006_at	660	248891_at
541	252412_at	581	251243_at	621	250005_at	661	248839_at
542	252337_at	582	251232_at	622	249997_at	662	248828_at
543	252326_at	583	251227_at	623	249894_at	663	248763_at
544	252192_at	584	251146_at	624	249861_at	664	248756_at
545	252098_at	585	251123_at	625	249836_at	665	248751_at
546	251989_at	586	251084_at	626	249829_at	666	248709_at
547	251985_at	587	251074_at	627	249817_at	667	248658_at
548	251962_at	588	251031_at	628	249785_at	668	248624_at
549	251954_at	589	251024_at	629	249777_at	669	248607_at
550	251935_at	590	251011_at	630	249774_at	670	248573_at
551	251902_at	591	250993_at	631	249701_at	671	248537_at
552	251869_at	592	250972_at	632	249677_at	672	248512_at
553	251860_at	593	250926_at	633	249645_at	673	248511_at
554	251855_at	594	250906_at	634	249610_at	674	248494_at
555	251852_at	595	250844_at	635	249582_at	675	248493_at
556	251846_at	596	250735_at	636	249569_at	676	248467_at
557	251840_at	597	250705_at	637	249546_at	677	248321_at
558	251834_at	598	250661_at	638	249521_at	678	248309_at
559	251815_at	599	250625_at	639	249510_at	679	248295_at

#	ID
680	248248_at
681	248246_at
682	248207_at
683	248195_at
684	248191_at
685	248190_at
686	248176_at
687	248155_at
688	248139_at
689	248126_at
690	248105_at
691	248083_at
692	248064_at
693	248062_at
694	248028_at
695	247987_at
696	247931_at
697	247923_at
698	247921_at
699	247910_at
700	247899_at
701	247891_at
702	247858_at
703	247817_at
704	247791_at
705	247787_at
706	247770_at
707	247766_at
708	247760_at
709	247724_at
710	247694_at
711	247692_s_at
712	247689_at
713	247651_at
714	247554_at
715	247544_at
716	247340_at
717	247328_at
718	247320_at
719	247318_at

#	ID
720	247295_at
721	247266_at
722	247232_at
723	247222_at
724	247193_at
725	247192_at
726	247097_at
727	247077_at
728	247069_at
729	247055_at
730	247042_at
731	247037_at
732	247025_at
733	247006_at
734	247000_at
735	246997_at
736	246976_s_at
737	246949_at
738	246920_at
739	246895_at
740	246837_at
741	246784_at
742	246783_at
743	246756_at
744	246744_at
745	246701_at
746	246700_at
747	246603_at
748	246554_at
749	246550_at
750	246548_at
751	246547_at
752	246540_at
753	246523_at
754	246522_at
755	246486_at
756	246439_at
757	246421_at
758	246403_at
759	246334_at

#	ID
760	246310_at
761	246304_at
762	246284_at
763	246249_at
764	246199_at
765	246154_at
766	246076_at
767	246043_at
768	245936_at
769	245877_at
770	245806_at
771	245775_at
772	245745_at
773	245734_at
774	245730_at
775	245724_at
776	245684_at
777	245642_at
778	245627_at
779	245619_at
780	245601_at
781	245435_at
782	245407_at
783	245404_at
784	245359_at
785	245348_at
786	245347_at
787	245340_at
788	245331_at
789	245319_at
790	245276_at
791	245270_at
792	245242_at
793	245218_s_at
794	245207_at
795	245195_at
796	245164_at
797	245152_at
798	245101_at
799	245094_at

#	ID
800	244996_at

Appendix B

Future directions: stochastic emulation

Abstract *This appendix provides a brief discussion of* stochastic emulation*, and illustrates ideas with a few simple examples. In both emulation and bootstrapping, we seek to approximate data generating processes (DGPs) using statistical models. However, in the emulation case, the "observed data" are simulated from a mathematical model. We note that many of the challenges associated with emulation (high dimensional feature spaces, large numbers of observations) are currently being tackled in the closely related context of simulation-based Bayesian inference (so-called "approximate Bayesian computation"). We draw some connections between these currently separate areas, which we believe may provide interesting directions for future research.*

B.1 Introduction

In Section 6.4 of the previous chapter, we illustrated a two-step approach for estimating the parameters of ODE systems. We demonstrated that this approach can provide significant computational savings relative to simulation-based procedures. This was largely because the two-step approach avoided having to solve the ODE numerically. Finding numerical solutions to ODE systems represents a significant computational bottleneck for simulation-based estimation procedures.

As we shall now discuss, *emulation* is a statistical learning approach that seeks to address this and similar problems. The goal is to reduce the computational cost of running an expensive computer program (such as a numerical integration routine) by approximating the relationship between the program's inputs and its outputs.

B.1.1 Deterministic emulation

We illustrate ideas in the deterministic case with a simple example. Consider the *repressilator* system of ODEs (Elowitz and Leibler, 2000), which was originally used to describe a synthetic gene network (and is now frequently considered in the literature as a toy model),

$$\begin{aligned} \frac{dm_1}{dt} &= -m_1 + \frac{\alpha}{1+p_3^n} + \alpha_0, & \frac{dp_1}{dt} &= -\beta(p_1 - m_1), \\ \frac{dm_2}{dt} &= -m_2 + \frac{\alpha}{1+p_1^n} + \alpha_0, & \frac{dp_2}{dt} &= -\beta(p_2 - m_2), \\ \frac{dm_3}{dt} &= -m_3 + \frac{\alpha}{1+p_2^n} + \alpha_0, & \frac{dp_3}{dt} &= -\beta(p_3 - m_3). \end{aligned} \tag{B.1}$$

Here, the p_i are protein concentrations, and the m_i are their corresponding mRNA concentrations. The four constants $(\beta, n, \alpha, \alpha_0)$ are parameters of the system. We assume that the α_0 and α parameters are known, and are respectively equal to 1 and 1,000. We further assume that the initial conditions are given, with $[m_1, p_1, m_2, p_2, m_3, p_3] = [0, 2, 0, 1, 0, 3]$ at time $t = 0$.

We suppose that we obtain a single observation, m_1^{obs}, of m_1 at time $t = 10$, and that we wish to use this in order to estimate the model's parameters. A simulation-based estimation routine would proceed by searching through a large number of different values for the β and n parameters. We denote by $(\beta^{(i)}, n^{(i)})$ the i^{th} pair of parameter values considered by the estimation routine, and by $m_1^{(i)}$ the simulated value of m_1 at time $t = 10$; that is, $m_1^{(i)} = m_1(t = 10; \beta^{(i)}, n^{(i)})$. The final estimates returned for β and n would be the pair, $(\beta^{(i)}, n^{(i)})$, for which the discrepancy between $m_1^{(i)}$ and m_1^{obs} is minimal (as quantified by some predefined error function, such as the squared difference).

Suppose that we were to stop our estimation procedure once it had searched through 500 pairs of values for β and n. Then we could plot the 500 simulated *outputs*, $m_1^{(i)}$, against their corresponding *inputs*, $(\beta^{(i)}, n^{(i)})$, as in Figure B.1a (note that here we actually sampled the parameter values $\beta^{(i)}$ and $n^{(i)}$ uniformly at random in the intervals $\beta^{(i)} \in [0, 10]$ and $n^{(i)} \in [0, 4]$, rather than using an optimisation procedure to suggest values). We can perhaps see from this figure that there is a pattern in the simulated output. This is better visualised by fitting an interpolating surface to the simulated outputs, as shown in Figure B.1b. There are many ways in which we could obtain such a surface, including: Gaussian process (Kennedy and O'Hagan, 2001); spline (Daughety and Turnquist, 1978); and artificial neural network (Anjum *et al.*, 1997) approaches. Regardless of how it is obtained, let us denote the interpolation model by $\hat{g}(\beta, n)$. For future (previously unseen) values, β_* and n_*, of the parameters, we may approximate $m_1(t = 10; \beta_*, n_*)$ by $\hat{g}(\beta_*, n_*)$. If we were now to resume our estimation procedure, then we could use the approximate model, $\hat{g}(\beta, n)$, instead of numerically solving the ODE system.

This general procedure of simulating a (large) number of times from a deterministic computer program, and then fitting an interpolating model, $\hat{g}$, in order to approximate the

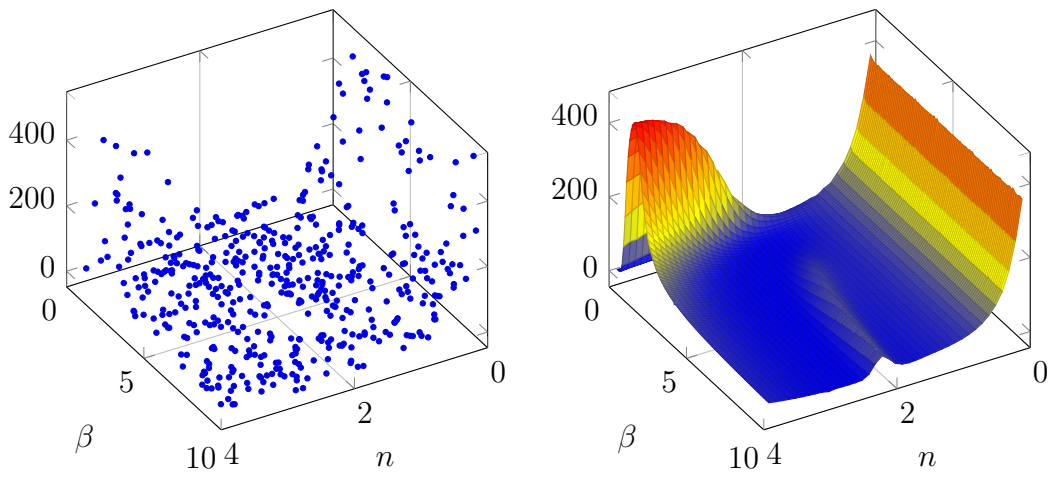


Figure B.1: (a) Simulated outputs, $m_1(t = 10)$, plotted against corresponding inputs, β and n; (b) a surface fitted to the points in (a) using the `griddata` function in Matlab (MATLAB, 2009)

input-output relationship is know as *emulation* (Conti *et al.*, 2009; Conti and O'Hagan, 2010; Liu and West, 2009). A closely related approach is *response surface modelling* (RSM), in which real experiments take the place of simulations, and the "parameters" correspond to variable experimental conditions (see Box and Wilson, 1951; Box and Draper, 2007).

One of the principal challenges of emulation is to ensure that the statistical model, $\hat{g}$, provides a good (or good enough) approximation to the model that generated the data (often referred to as the *simulator*). A judicious choice of *design points* (the selection of inputs that we use to generate the initial, simulated data set) can be helpful, and *latin hypercube sampling* (McKay *et al.*, 1979) is often employed in order to obtain a good coverage of the input space. Once we have trained $\hat{g}$, we may quantify its goodness of fit with reference to a validation data set, generated using the simulator (Bastos and O'Hagan, 2009). If we deem the fit to be poor, simulating more training data might improve matters (although, of course, this will add to the overall computational cost of training the emulator).

It is common to use emulation in *sensitivity analyses* (Saltelli *et al.*, 2000), which seek to quantify the rate of change of a model's outputs with respect to its inputs (see, for example Kennedy *et al.*, 2006). However, more elaborate applications have also been proposed, including the use of emulators to speed up inference procedures (Rasmussen, 2003).

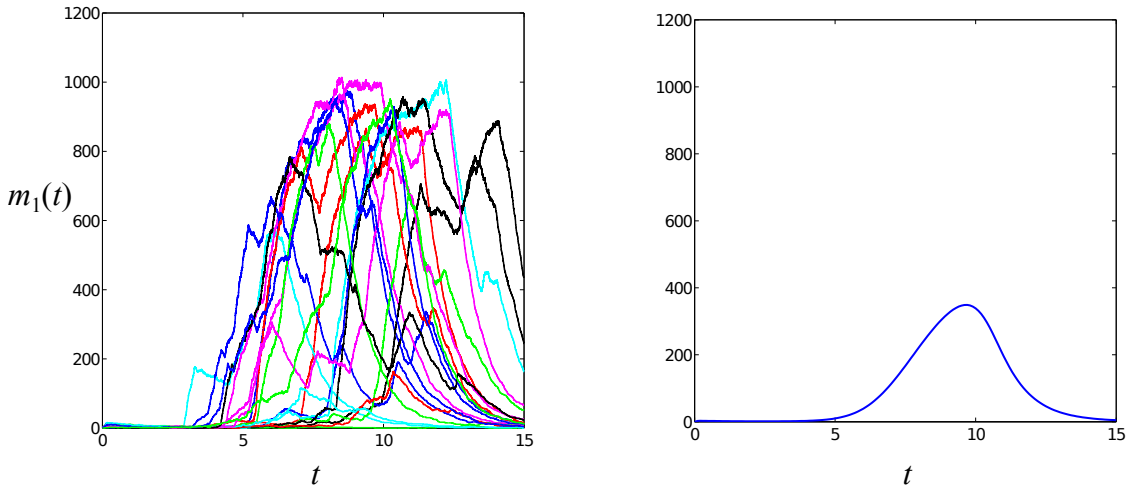
B.2 Stochastic emulation

Recently, there has been interest in extending emulation to *stochastic* models, which may also be computationally costly to simulate (Henderson *et al.*, 2009). The difficulty here lies in the fact that, for any given choice of the model's inputs (parameters), there is now a distribution of possible outputs. As we shall shortly discuss, this requires us to consider methods for density estimation in place of methods for interpolation.

To provide an example, we follow Toni *et al.* (2009) and transform the repressilator model of Equation (B.1) into the following set of reactions [1],

$$\begin{array}{rlll} \text{mRNA synthesis:} & \emptyset \rightarrow m_i & \text{with hazard} & \dfrac{\alpha}{1+p_j^n} + \alpha_0 \\ \text{mRNA degradation:} & m_i \rightarrow \emptyset & \text{with hazard} & m_i \\ \text{Protein synthesis:} & m_i \rightarrow m_i + p_i & \text{with hazard} & \beta m_i \\ \text{Protein degradation:} & p_i \rightarrow \emptyset & \text{with hazard} & \beta p_i \end{array} \qquad \text{(B.2)}$$

We simulate these reactions using the stochastic simulation algorithm (SSA) of Gillespie 1977 (see also Wilkinson, 2006). In Figure B.2a we show the results of simulating 20 times using the SSA, using randomly chosen values for n and β (and fixing α and α_0 as before). This should be compared to Figure B.2b, in which we have simulated the deterministic system.



(a) (b)

Figure B.2: Simulating the repressilator system when $n = 2.9627$ and $\beta = 0.8753$, in (a) stochastic, and (b) deterministic cases.

It is clear from Figure B.2a that we may no longer use interpolation in order to approximate the relationship between the inputs of our stochastic model and its output. Even for

[1] We refer to Wilkinson (2006) for background regarding stochastic modelling.

fixed input values ($n = 2.9627$ and $\beta = 0.8753$ in Figure B.2a), the stochastic nature of the model means that — in contrast to the deterministic case — there is not a single, fixed value for the model's output (e.g. $m_1(t)$ at time $t = 10$). For the sake of generality, we shall henceforth denote the model's output(s) by $\mathbf{y}$, and its inputs by $\mathbf{x}$.

How should we approach emulation in the stochastic case? For given inputs, $\mathbf{x}_*$, the stochastic model's outputs are described by the conditional distribution $p(\mathbf{y}|\mathbf{x} = \mathbf{x}_*)$. A stochastic emulator must therefore provide an approximation to this conditional distribution for any choice of $\mathbf{x}_*$. From this point, there would seem to be two ways in which to proceed, which we now discuss (Sections B.2.1 and B.2.2).

B.2.1 The parametric approach

Suppose that we were to take the mean of the 20 values of $m_1(t)$ at time $t = 10$ in Figure B.2a. Then, corresponding to this particular choice for β and n, we would have a single summary of the model's output, which we shall denote by ρ. Repeating this for many different values of β and n, we obtain a large number of triples, $(\beta^{(i)}, n^{(i)}, \rho^{(i)})$. We may then fit a regression model to these data to describe how ρ varies with β and n. More generally, ρ could be any summary (or combination of summaries) of the simulated data. If we were to assume a parametric model for the conditional density, say $p(\mathbf{y}|\mathbf{x}) = q(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x}))$, then we could take $\rho = \boldsymbol{\theta}(\mathbf{x})$ to be this model's parameters.

For example, this *regression approach* forms part of the procedure adopted by Henderson *et al.* (2009) in order to emulate a stochastic model of mitochondrial DNA deletions in *substantia nigra* neurons. We now very briefly distil some of the main ideas of this paper, with some simplifications for the sake of brevity. The authors construct a training data set by considering 250 different inputs, and — for each one — they simulate 1000 times[2] from their model. For each input, the authors calculate the sample mean and standard deviation of the corresponding simulated output. They then fit independent Gaussian process regression models to describe how these two summaries vary as a function of the inputs. It is assumed that the conditional distribution, $p(\mathbf{y}|\mathbf{x})$, is approximately univariate normal, and hence is completely specified by the two summaries. That is, for any given input, $\mathbf{x} = \mathbf{x}_*$, the authors may use their GP regression models in order to predict the mean and standard deviation, μ_* and σ_*, of the corresponding output. The conditional density, $p(\mathbf{y}|\mathbf{x})$, is then approximated as $\mathcal{N}(\mu_*, \sigma_*)$.

The main limitation of this approach is immediately apparent: what if there is an $\mathbf{x}$ for which a Gaussian is a poor model for $p(\mathbf{y}|\mathbf{x})$? Indeed, any choice of parametric model, $q(\mathbf{y}|\boldsymbol{\theta}(\mathbf{x}))$, places constraints on the behaviour that may be described, and these may prove unreasonable for (at least) some choices of $\mathbf{x}$. *Stochastic bifurcations* (where, for example, there is some critical value of $\mathbf{x}$ at which $p(\mathbf{y}|\mathbf{x})$ switches from a unimodal to a

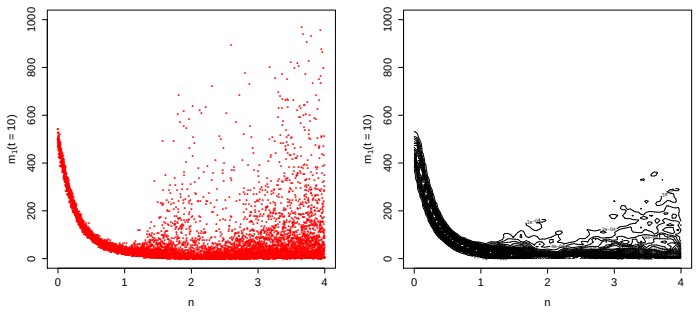
[2] In fact, for each input, the authors group their simulations into 40 sets of 25. This is due to a feature of their stochastic model, which permits null outcomes. We refer to Henderson *et al.* (2009) for full details of their approach.

bimodal distribution) might provide particular difficulties for such parametric approaches (see Song *et al.*, 2010, for illustrations of stochastic bifurcations in cellular networks). The principal difficulty, however, is that it is very difficult to decide upon a parametric form that is appropriate for all choices of $\mathbf{x}$.

B.2.2 The nonparametric approach

Instead of assuming a fixed parametric model, we may instead consider nonparametric methods for estimating $p(\mathbf{y}|\mathbf{x})$. We illustrate using *kernel density estimators*, and refer to Hall *et al.* (2004) and references therein for details of these approaches.

To provide an example, we return to the stochastic repressilator of Equation (B.2). As previously, we assume that the α_0 and α parameters are known, and are respectively equal to 1 and 1,000. Additionally, we now assume that β is known and equal to 5. We sample 10,000 values for n uniformly at random from the interval $[0, 4]$, and — for each one — we simulate from Equation (B.2) using the stochastic simulation algorithm. For simplicity, we assume that we are again interested in the value of $m_1(t)$ at time $t = 10$. The simulated input-output pairs are plotted in Figure B.3a. We apply kernel density estimation (KDE) to these data, using of the `np` package in `R` (Hayfield and Racine, 2008). We use a second order Gaussian kernel, whose bandwidth is estimated using a maximum likelihood cross-validation approach (see Li and Racine, 2007, Chapter 5). We approximate both $p(m_1(t = 10), n)$ (see Figure B.3b) and also $p(m_1(t = 10)|n)$ (for a variety of choices for n) using kernel density estimation. When approximating $p(m_1(t = 10)|n)$, we deliberately chose values of n that did not appear amongst our original 10,000 sampled values; namely, $n = 0.88, 1.24, 2.17$ and 3.66. We then assess the quality of these approximate conditionals by simulating 100 times for each value of n. Both the kernel density estimates and the distributions of simulated outputs are plotted in Figure B.4.



(a) (b)

Figure B.3: (a) Data generated by simulating from the stochastic repressilator of Equation (B.2). (b) Contour lines illustrating the kernel density estimate of $p(m_1(t = 10), n)$

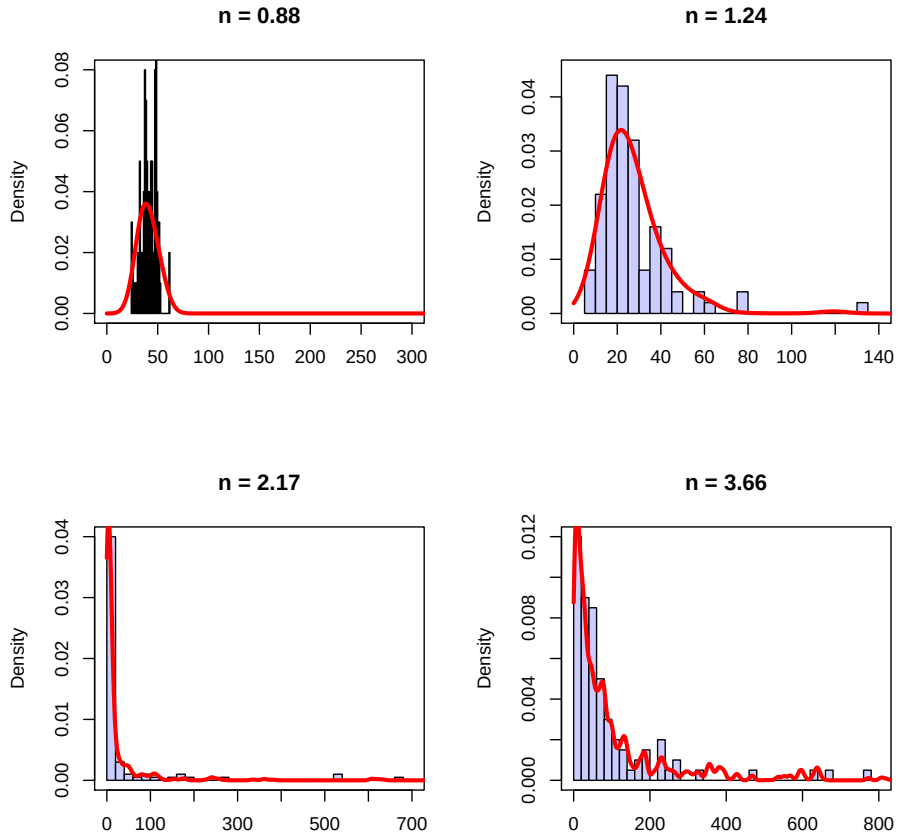


Figure B.4: Plots showing both kernel density estimates of $p(m_1(t = 10)|n)$ (red curves) and samples drawn from the true distribution by simulation (blue histograms). The title of each plot provides the relevant value for n. Note that each of these plots can be considered as a vertical "slice" through Figure B.3b, taken at the values of n given in the plot titles.

Figure B.4 shows that the kernel density estimates of $p(m_1(t = 10)|n)$ are generally very good. For the smallest value, $n = 0.88$, we have the worst fit, with the estimated density being broader and rather less peaked than the true distribution. Depending on the task of interest, however, even this approximation may be adequate. Perhaps the most important feature of these plots, however, is that they show that the shape of the conditional distribution changes with n. A Gaussian certainly would not provide a good approximation to $p(m_1(t = 10)|n)$ for all values of n (although it does appear that a log-normal distribution might provide a reasonable fit).

B.2.2.1 Emulation versus inference

In the previous section, we illustrated the use of kernel density estimators for approximating the conditional distributions $p(\mathbf{y}|\mathbf{x})$. In the caption to Figure B.4, we explained that these conditionals may be considered as vertical "slices" taken through the plots shown in Figure B.3. We could employ similar approaches in order to take horizontal "slices" instead, and hence approximate $p(\mathbf{x}|\mathbf{y})$. Since $\mathbf{y}$ denotes the outputs of our simulator, it

will often represent quantities that can be measured experimentally (e.g. in our repressilator example, the number of molecules of a species of mRNA). We would hence be able to approximate $p(\mathbf{x}|\mathbf{y} = \mathbf{y}^{\text{obs}})$, where $\mathbf{y}^{\text{obs}}$ represents an experimentally obtained observation of $\mathbf{y}$. In the particular case where the distribution from which we sampled our design points may be assumed to represent a prior distribution for the inputs, this approximation to $p(\mathbf{x}|\mathbf{y} = \mathbf{y}^{\text{obs}})$ represents an approximation to the usual Bayesian posterior.

This approach for approximating the posterior is similar in spirit to the one taken in approximate Bayesian computation (ABC) procedures (Marjoram *et al.*, 2003; Sisson *et al.*, 2007; Ratmann *et al.*, 2009). In the simplest such procedure (ABC rejection), a sample is drawn from the prior for the inputs, $p(\mathbf{x})$, and is "accepted" if the resulting simulator output is sufficiently close to $\mathbf{y}^{\text{obs}}$. Here, "closeness" is quantified by a pre-specified distance function, d, and "sufficiently close" means that the distance between the simulated output and $\mathbf{y}^{\text{obs}}$ falls below some threshold level, ϵ. Repeating a large number of times generates a collection of "accepted" samples. As explained in Marjoram *et al.* (2003), these represent samples from the distribution, $p(\mathbf{x}|d(\mathbf{y}, \mathbf{y}^{\text{obs}}) < \epsilon)$, which is taken as an approximation to the posterior $p(\mathbf{x}|\mathbf{y} = \mathbf{y}^{\text{obs}})$. This is very similar to the procedure outlined in the previous paragraph, with the only difference being the method by which the conditional distribution, $p(\mathbf{x}|\mathbf{y} = \mathbf{y}^{\text{obs}})$, is approximated.

B.2.2.2 Alternative nonparametric methods

The kernel density estimation approach is relatively simple, and seems to provide reasonable results. However, for large data sets (such as the 10,000 simulated observations that we have in our example), bandwidth estimation can be slow. Additionally, the final model is rather unwieldy, since it comprises a weighted sum of M kernel functions, where M is the number of observations. An alternative approach is to fit a Gaussian mixture model (GMM) to data sampled from the joint distribution, $p(\mathbf{x}, \mathbf{y})$. Choosing an *optimal* number of components is a challenging problem, but there are a number of possibilities, including: optimising the Bayesian information criterion (Fraley and Raftery, 2002); variational Bayesian approaches (Teschendorff *et al.*, 2005); and approximate procedures derived from a nonparametric Bayesian (Dirichlet process) standpoint (Heller and Ghahramani, 2005; Daumé III, 2009). "Fully" Bayesian nonparametric approaches are also possible (Müller *et al.*, 1996; Dunson *et al.*, 2007; Jara, 2007); however, it seems likely that the computational costs of applying such methods would limit their practicality. Regardless of how we choose the number of components, once we have fitted the GMM — which we shall denote by $q(\mathbf{x}, \mathbf{y})$ — it is possible to calculate the conditionals, $q(\mathbf{y}|\mathbf{x})$, analytically (see Müller *et al.*, 1996). We may hence use these to approximate $p(\mathbf{y}|\mathbf{x})$. The principal advantage of such an approach is that, in contrast to the kernel density approaches, the final model will typically have far fewer components than the number of observations. In light of Section B.2.2.1 above, an additional advantage is that we approximate $p(\mathbf{x}|\mathbf{y})$ for any given $\mathbf{y}$ just as easily as we may approximate $p(\mathbf{y}|\mathbf{x})$.

B.3 Discussion

The principal challenge for stochastic emulation is approximating the (usually) high-dimensional distribution, $p(\mathbf{y}|\mathbf{x})$. Assuming a parametric model for this distribution and determining how its parameters, $\boldsymbol{\theta}$, vary as a function of $\mathbf{x}$ (as in Section B.2.1), may provide a method for simplifying this problem. Essentially, $\boldsymbol{\theta}(\mathbf{x})$ is being used as a (lower-dimensional) summary for $\mathbf{y}(\mathbf{x})$. As discussed, however, the limitation of such an approach is that it places strong constraints on the possible behaviours that may be described. If we wish to use nonparametric methods instead, one way to reduce the dimensionality of $\mathbf{y}$ might be to choose a summary statistic, $\rho(\mathbf{y})$, and to use this in place of $\mathbf{y}$. We may consider, for example, that in Section B.2, we used the value of $m_1(t)$ at time $t = 10$ as a summary for the trajectories illustrated in Figure B.2a (which are themselves a summary of the output of the stochastic repressilator model given in Equation B.2). Summarising high-dimensional simulator outputs using low dimensional statistics is one of the "tricks" that is often employed by ABC procedures (Marjoram *et al.*, 2003), for much the same reasons as they might be used here. Given the similarities between stochastic emulation and approximate Bayesian computation (and the common challenges faced by both) we believe that an interesting direction for future research would be to determine how recent innovations for ABC (e.g. Blum and Francois, 2010) may be applied to stochastic emulation problems.

References

Aaronson, D. S. and Horvath, C. M. (2002). A road map for those who don't know JAK-STAT. *Science*, **296**(5573), 1653–1655. (page 103.)

Abeel, T., Helleputte, T., de Peer, Y. V., Dupont, P., and Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, **26**(3), 392–8. (pages 27, 28, 31, and 47.)

Anjum, M., Tasadduq, I., and Al-Sultan, K. (1997). Response surface methodology: A neural network approach. *Eur J Oper Res*, **101**(1), 65–73. (page 127.)

Asquith, B., Mosley, A. J., Heaps, A., Tanaka, Y., Taylor, G. P., McLean, A. R., and Bangham, C. R. M. (2005). Quantification of the virus-host interaction in human T lymphotropic virus I infection. *Retrovirology*, **2**, 75. (page 51.)

Baggerly, K. A., Morris, J. S., and Coombes, K. R. (2004). Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics*, **20**(5), 777–85. (pages 8, 52, 57, 115, and 116.)

Bangham, C. R. M. (2000). HTLV-1 infections. *J Clin Pathol*, **53**(8), 581–6. (page 50.)

Bangham, C. R. M., Nightingale, S., Cruickshank, J. K., and Daenke, S. (1989). PCR Analysis of DNA from Multiple Sclerosis Patients for the Presence of HTLV-I. *Science*, **246**(4931), 821–821. (page 63.)

Barabasi, A. and Oltvai, Z. (2004). Network biology: Understanding the cell's functional organization. *Nat Rev Genet*, **5**(2), 101–U15. (page 78.)

Barber, D. and Williams, C. K. I. (1997). Gaussian processes for Bayesian classification via hybrid Monte Carlo. In *Advances in Neural Information Processing Systems 9*, pages 340–346. MIT Press. (page 71.)

Barla, A., Jurman, G., Riccadonna, S., Merler, S., Chierici, M., and Furlanello, C. (2008). Machine learning methods for predictive proteomics. *Brief Bioinformatics*, **9**(2), 119–28. (page 9.)

Bastos, L. S. and O'Hagan, A. (2009). Diagnostics for Gaussian Process Emulators. *Technometrics*, **51**(4), 425–438. (page 128.)

Bayarri, M. J. and Berger, J. O. (1999). Quantifying surprise in the data and model verification. *Bayesian statistics 6*, pages 53–82. (page 21.)

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Roy. Statist. Soc. Ser. B*, **57**(1), 289–300. (page 10.)

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. Wiley. (page 71.)

Biegler, L., Damiano, J., and Blau, G. (1986). Nonlinear parameter-estimation - a case-study comparison. *Aiche J*, **32**(1), 29–45. (page 97.)

Blum, M. G. B. and Francois, O. (2010). Non-linear regression models for Approximate Bayesian Computation. *Stat Comput*, **20**(1), 63–73. (page 134.)

Bollback, J. P. (2005). Posterior mapping and posterior predictive distributions. In *Statistical Methods in Molecular Evolution*, pages 439–462. Springer. (page 22.)

Box, G. E. P. (1980). Sampling and bayes' inference in scientific modelling and robustness. *Journal of the Royal Statistical Society. Series A (General)*, **143**(4), 383–430. (page 21.)

Box, G. E. P. and Draper, N. R. (2007). *Response Surfaces, Mixtures, and Ridge Analyses*. Wiley, Hoboken, NJ, second edition. (page 128.)

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *J. Roy. Statist. Soc. Ser. B*, **13**, 1–38. (page 128.)

Boyle, P. and Frean, M. (2005). Dependent gaussian processes. In *Advances in Neural Information Processing Systems 17*, pages 217–224. (page 118.)

Broadhurst, D. I. and Kell, D. B. (2006). Statistical strategies for avoiding false discoveries in metabolomics and related experiments. *Metabolomics*, **2**(4), 171–196. (page 9.)

Brown, J. K. M. (1994). Bootstrap hypothesis tests for evolutionary trees and other dendrograms. *P Natl Acad Sci Usa*, **91**(25), 12293–12297. (pages 87 and 92.)

Brunel, N. (2008). Parameter estimation of ODE's via nonparametric estimators. *Electronic Journal of Statistics*, **2**, 1242–1267. (pages 96 and 97.)

Bühlmann, P. (2002). Bootstraps for time series. *Stat Sci*, **17**(1), 52–72. (page 65.)

Bullard, J. H., Purdom, E., Hansen, K. D., and Dudoit, S. (2010). Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, **11**, 94. (page 26.)

Butte, A., Tamayo, P., Slonim, D., Golub, T., and Kohane, I. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *P Natl Acad Sci Usa*, **97**(22), 12182–12186. (pages 11, 67, and 79.)

Calderhead, B., Girolami, M., and Lawrence, N. D. (2009). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 217–224. MIT Press. (pages 113 and 117.)

Carlin, B. P. (1999). Discussion on the paper by Bayarri and Berger. *Bayesian statistics 6*, pages 53–82. (page 22.)

Christley, S., Nie, Q., and Xie, X. (2009). Incorporating existing network information into gene network inference. *PLoS ONE*, **4**(8), e6799. (page 79.)

Churchill, G. A. (2004). Using ANOVA to analyze microarray data. *BioTechniques*, **37**(2), 173–5, 177. (page 53.)

Coleman, T. and Li, Y. (1996). An interior trust region approach for nonlinear minimization subject to bounds. *Siam J Optimiz*, **6**(2), 418–445. (page 99.)

Coleman, T. F. and Li, Y. (1994). On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical Programming*, **67**, 189–224. 10.1007/BF01582221. (page 99.)

Constans, A. (2006). Serum proteomics scrutinized. *The Scientist*, **20**(5), 65–66. (page 52.)

Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *J. Statist. Plann. Inference*, **140**(3), 640–651. (pages 118 and 128.)

Conti, S., Gosling, J. P., Oakley, J. E., and O'Hagan, A. (2009). Gaussian process emulation of dynamic computer codes. *Biometrika*, **96**(3), 663–676. (page 128.)

Coombes, K. R., Fritsche, H. A., Clarke, C., Chen, J.-N., Baggerly, K. A., Morris, J. S., Xiao, L.-C., Hung, M.-C., and Kuerer, H. M. (2003). Quality control and peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption and ionization. *Clin Chem*, **49**(10), 1615–23. (pages 53 and 57.)

Daughety, A. F. and Turnquist, M. A. (1978). Simulation optimization using response surfaces based on spline approximations. In *WSC '78: Proceedings of the 10th conference on Winter simulation*, pages 183–193, Piscataway, NJ, USA. IEEE Press. (page 127.)

Daumé III, H. (2009). Fast search for dirichlet process mixture models. *arXiv*, **cs.LG**. Published in: AIStats 2007. (page 133.)

Davison, A. and Hinkley, D. V. (1999). *Bootstrap methods and their application*. Cambridge University Press. (page 65.)

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol*, **9**(1), 67–103. (page 78.)

de Silva, E. and Stumpf, M. (2005). Complex networks and simple models in biology. *J Roy Soc Interface*, **2**(5), 419–430. (page 78.)

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, **26**(3), 297–302. (page 31.)

Dormand, J. R. and Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *J. Comput. Appl. Math.*, **6**(1), 19–26. (page 96.)

Dunne, K., Cunningham, P., and Azuaje, F. (2002). Solutions to instability problems with sequential wrapper-based approaches to feature selection. Technical report, Department of Computer Science, Trinity College, Dublin. Technical report. (page 31.)

Dunson, D. B., Pillai, N., and Park, J.-H. (2007). Bayesian density regression. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **69**(2), 163–183. (page 133.)

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, **7**(1), 1–26. (pages 12 and 15.)

Efron, B. (2003). Second thoughts on the bootstrap. *Statistical Science*, **18**(2), 135–140. Silver Anniversary of the Bootstrap. (page 13.)

Efron, B. (2004). Large-scale simultaneous hypothesis testing: The choice of a null hypothesis. *J Am Stat Assoc*, **99**(465), 96–104. (pages 80 and 81.)

Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *J Am Stat Assoc*, **92**(438), 548–560. (page 16.)

Efron, B. and Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Chapman and Hall/CRC. (pages 12, 13, 15, 16, 65, and 115.)

Efron, B., Halloran, E., and Holmes, S. (1996). Bootstrap confidence levels for phylogenetic trees. *P Natl Acad Sci Usa*, **93**(23), 13429–34. (page 15.)

Efron, B., Tibshirani, R., Storey, J., and Tusher, V. (2001). Empirical bayes analysis of a microarray experiment. *J Am Stat Assoc*, **96**(456), 1151–1160. (pages 9 and 26.)

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Ann Stat*, **32**(2), 407–451. (pages 32, 43, 86, and 90.)

Ein-Dor, L., Kela, I., Getz, G., Givol, D., and Domany, E. (2005). Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, **21**(2), 171–8. (page 27.)

Ein-Dor, L., Zuk, O., and Domany, E. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc Natl Acad Sci USA*, **103**(15), 5923–8. (pages 8 and 27.)

Elowitz, M. B. and Leibler, S. (2000). A synthetic oscillatory network of transcriptional regulators. *Nature*, **403**(6767), 335–8. (page 127.)

Feller, W. (1968). *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley. (page 30.)

Felsenstein, J. (1985). Confidence limits on phylogenies: An approach using the bootstrap. *Evolution*, **39**(4), 783–791. (pages 15, 87, 92, and 115.)

Fisher, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd. (page 10.)

Foell, D. and Roth, J. (2004). Proinflammatory S100 proteins in arthritis and autoimmune disease. *Arthritis Rheum*, **50**(12), 3762–71. (pages 62 and 116.)

Fox, R. and Dimmic, M. (2006). A two-sample Bayesian t-test for microarray data. *BMC Bioinformatics*, **7**, 126. (page 10.)

Fraley, C. and Raftery, A. (2002). Model-based clustering, discriminant analysis, and density estimation. *J Am Stat Assoc*, **97**(458), 611–631. (page 133.)

Freund, J. E. (1971). *Mathematical statistics*. Prentice-Hall, Englewood Cliffs, N.J., 2nd ed. edition. (page 86.)

Friedel, C. C., Krumsiek, J., and Zimmer, R. (2009). Bootstrapping the interactome: unsupervised identification of protein complexes in yeast. *J Comput Biol*, **16**(8), 971–87. (page 15.)

Friedman, J., Hastie, T., Hoefling, H., and Tibshirani, R. (2007). Pathwise coordinate optimization. *Ann Appl Stat*, **1**(2), 302–332. (page 28.)

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33**(1), 1–22. (pages 28, 33, and 43.)

Gao, P., Honkela, A., Rattray, M., and Lawrence, N. D. (2008). Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics*, **24**(16), i70–75. (page 67.)

Gelman, A. (2004). *Bayesian data analysis*. Chapman and Hall/CRC. (pages 19, 20, 21, and 71.)

Gelman, A., Meng, X., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Stat Sinica*, **6**(4), 733–760. (pages 14, 20, 21, 22, and 97.)

Genkin, A., Lewis, D. D., and Madigan, D. (2007). Large-scale Bayesian logistic regression for text categorization. *Technometrics*, **49**(3), 291–304. (page 33.)

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, **81**(25), 2340–2361. (pages 118 and 129.)

Girard, A. (2004). Approximate methods for propagation of uncertainty with Gaussian process models (Doctoral dissertation, University of Glasgow). (page 109.)

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, 3rd edition. (page 77.)

Goon, P. K. C., Igakura, T., Hanon, E., Mosley, A. J., Asquith, B., Gould, K. G., Taylor, G. P., Weber, J. N., and Bangham, C. R. M. (2003). High circulating frequencies of tumor necrosis factor alpha- and interleukin-2-secreting human T-lymphotropic virus type 1 (HTLV-1)-specific CD4+ T cells in patients with HTLV-1-associated neurological disease. *J Virol*, **77**(17), 9716–22. (page 51.)

Grigorov, M. G. (2006). Global dynamics of biological systems from time-resolved omics experiments. *Bioinformatics*, **22**(12), 1424–30. (page 65.)

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., and Sethna, J. P. (2007). Universally sloppy parameter sensitivities in systems biology models. *PLoS Comput Biol*, **3**(10), 1871–1878. (page 11.)

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*. (page 27.)

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, **46**(1-3), 389–422. (page 28.)

Hall, P., Racine, J., and Li, Q. (2004). Cross-validation and the estimation of conditional probability densities. *J Am Stat Assoc*, **99**(468), 1015–1026. (page 131.)

Hand, D. J. (2008). Breast cancer diagnosis from proteomic mass spectrometry data: A comparative evaluation. *Statistical applications in genetics and molecular biology*, **7**(2), 15. (pages 52 and 57.)

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, 2nd edition. (pages 19 and 33.)

Hayfield, T. and Racine, J. S. (2008). Nonparametric econometrics: The np package. *J Stat Softw*, **27**(5), 1–32. (page 131.)

He, Z. and Yu, W. (2010). Stable feature selection for biomarker discovery. *arXiv*, **cs.CE**. (pages 27, 31, and 90.)

Heller, K. and Ghahramani, Z. (2005). Bayesian hierarchical clustering. *Proceedings of the 22nd international conference on Machine learning*. (page 133.)

Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., and Wilkinson, D. J. (2009). Bayesian emulation and calibration of a stochastic computer model of mitochondrial dna deletions in substantia nigra neurons. *J Am Stat Assoc*, **104**(485), 76–87. (pages 118, 129, and 130.)

Hoerl, A. E. (1962). Application of ridge analysis to regression problems. *Chem Eng Prog*, **58**, 54–59. (page 33.)

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scand J Stat*, **6**(2), 65–70. (page 10.)

Horvath, C. M. (2000). STAT proteins and transcriptional responses to extracellular signals. *Trends in Biochemical Sciences*, **25**(10), 496–502. (page 103.)

Hotelling, H. (1953). New light on the correlation coefficient and its transforms. *J. Roy. Statist. Soc. Ser. B.*, **15**, 193–225; discussion, 225–232. (page 81.)

Imoto, S., Kim, S., Goto, T., Miyano, S., Aburatani, S., Tashiro, K., and Kuhara, S. (2003). Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *J Bioinform Comput Biol*, **1**(2), 231–52. (page 84.)

Imoto, S., Higuchi, T., Kim, S., Jeong, E., and Miyano, S. (2004). Residual bootstrapping and median filtering for robust estimation of gene networks from microarray data. In *CMSB*, pages 149–160. (page 15.)

Issaq, H. J., Veenstra, T. D., Conrads, T. P., and Felschow, D. (2002). The SELDI-TOF MS approach to proteomics: protein profiling and biomarker identification. *Biochem Biophys Res Commun*, **292**(3), 587–92. (pages 51 and 52.)

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, **37**, 547–579. (page 29.)

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, **11**(2), 37–50. (page 29.)

Jara, A. (2007). Applied Bayesian Non- and Semi-parametric Inference using DPpackage. *Rnews*, **7**(3), 17–26. (page 133.)

Jasra, A., Stephens, D. A., and Holmes, C. C. (2007). On population-based simulation for static inference. *Stat Comput*, **17**(3), 263–279. (page 113.)

Ji, X. and Xu, Y. (2006). libSRES: a C library for stochastic ranking evolution strategy for parameter estimation. *Bioinformatics*, **22**(1), 124–126. (page 104.)

Jost, C. and Ellner, S. (2000). Testing for predator dependence in predator-prey dynamics: a non-parametric approach. *P Roy Soc Lond B Bio*, **267**(1453), 1611–1620. (page 97.)

Kalousis, A., Prados, J., and Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems*. (pages 30 and 31.)

Kelly, S. E., Jones, D. B., and Fleming, S. (1989). Calgranulin expression in inflammatory dermatoses. *J Pathol*, **159**(1), 17–21. (page 62.)

Kelly, W. P. and Stumpf, M. P. H. (2008). Protein–protein interactions: from global to local analyses. *Current Opinion in Biotechnology*. (page 78.)

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, **63**(3), 425–464. (pages 118 and 127.)

Kennedy, M. C., Anderson, C. W., Conti, S., and O'Hagan, A. (2006). Case studies in Gaussian process modelling of computer codes. *Reliab Eng Syst Safe*, **91**(10-11), 1301–1309. (page 128.)

Kent, J. T. (2010). Discussion on the paper by Meinshausen and Bühlmann. *J R Stat Soc B*, **72**(4), 458. (page 36.)

Kerr, M. K. and Churchill, G. A. (2001). Bootstrapping cluster analysis: Assessing the reliability of conclusions from microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, **98**(16), 8961–8965. (pages 15 and 67.)

Kim, R. D. and Park, P. J. (2004). Improving identification of differentially expressed genes in microarray studies using information from public databases. *Genome Biol*, **5**(9), R70. (page 10.)

Kirk, P. D., Toni, T., and Stumpf, M. P. (2008). Parameter Inference for Biochemical Systems that undergo a Hopf Bifurcation. *Biophys. J.*, **Preprint**. (pages 11 and 97.)

Kirk, P. D. W., Lewin, A. M., and Stumpf, M. P. H. (2010). Discussion on the paper by Meinshausen and Bühlmann. *J R Stat Soc B*, **72**(4), 456–457. (pages 28 and 36.)

Klebanov, L. and Yakovlev, A. (2007). How high is the level of technical noise in microarray data? *Biol Direct*, **2**, 9. (page 11.)

Koh, K., Kim, S.-J., and Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, **8**, 1519–1555. (page 33.)

Kohavi, R. and John, G. (1997). Wrappers for feature subset selection. *Artif Intell*, **97**(1-2), 273–324. (page 27.)

Kuncheva, L. (2007). A stability index for feature selection. In *Proceedings of the 25th International Multi-Conference on Artificial Intelligence and Applications*. (page 30.)

Lai, Y. (2008). Genome-wide co-expression based prediction of differential expressions. *Bioinformatics*, **24**(5), 666–73. (page 11.)

Lander, A. D. (2010). The edges of understanding. *BMC Biol*, **8**, 40. (page 87.)

Lauritzen, S. L. (1996). *Graphical models*. Clarendon Press, Oxford. (page 80.)

Lawrence, N. D., Sanguinetti, G., and Rattray, M. (2007). Modelling transcriptional regulation using Gaussian Processes. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 785–792. MIT Press, Cambridge, MA. (page 67.)

Lèbre, S. (2009). Inferring dynamic genetic networks with low order independencies. *Statistical applications in genetics and molecular biology*, **8**(1), Article 9. (pages 11 and 93.)

Lee, M. L., Kuo, F. C., Whitmore, G. A., and Sklar, J. (2000). Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *P Natl Acad Sci Usa*, **97**(18), 9834–9. (page 11.)

Lewin, A., Bochkina, N., and Richardson, S. (2007). Fully bayesian mixture model for differential gene expression: Simulations and model checks. *Statistical applications in genetics and molecular biology*, **6**, 36. (page 22.)

Li, Q. and Racine, J. S. (2007). *Nonparametric econometrics: theory and practice*. Princeton University Press. (page 131.)

Lindley, D. V. (1999). Discussion on the paper by Bayarri and Berger. *Bayesian statistics 6*, pages 53–82. (page 22.)

Liu, F. and West, M. (2009). A Dynamic Modelling Strategy for Bayesian Computer Model Emulation. *Bayesian Analysis*, **4**(2), 393–411. (page 128.)

Liu, Q., Lin, K. K., Andersen, B., Smyth, P., and Ihler, A. (2010). Estimating replicate time shifts using Gaussian process regression. *Bioinformatics*, **26**(6), 770–6. (page 67.)

Lotka, A. J. (1920). Analytical note on certain rhythmic relations in organic systems. *P Natl Acad Sci Usa*, **6**(7), 410–5. (page 98.)

Ma, H. and Zeng, A.-P. (2003). Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. *Bioinformatics*, **19**(2), 270–7. (page 78.)

MacGregor, G., Gray, R. D., Hilliard, T. N., Imrie, M., Boyd, A. C., Alton, E. W. F. W., Bush, A., Davies, J. C., Innes, J. A., Porteous, D. J., and Greening, A. P. (2008). Biomarkers for cystic fibrosis lung disease: Application of SELDI-TOF mass spectrometry to BAL fluid. *J Cyst Fibros*, **7**(5), 352–358. (page 52.)

Manicourt, D., Brauman, H., and Orloff, S. (1978). Plasma and urinary levels of $\beta 2$ microglobulin in rheumatoid arthritis. *Ann Rheum Dis*, **37**(4), 328–32. (page 62.)

Mann, H. and Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, **18**(1), 50–60. (page 54.)

Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., and Califano, A. (2006). Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, **7**, S7. (pages 11 and 79.)

Marjoram, P., Molitor, J., Plagnol, V., and Tavare, S. (2003). Markov chain monte carlo without likelihoods. *Proc Natl Acad Sci USA*, **100**(26), 15324–8. (pages 118, 133, and 134.)

Marshall, E. (2004). Getting the noise out of gene arrays. *Science*, **306**(5696), 630–1. (pages 9 and 11.)

Marshall, E. C. and Spiegelhalter, D. J. (2003). Approximate cross-validatory predictive checks in disease mapping models. *Stat Med*, **22**(10), 1649–60. (page 22.)

Marshall, E. C. and Spiegelhalter, D. J. (2007). Identifying outliers in bayesian hierarchical models: a simulation-based approach. *Bayesian Analysis*, **2**(2), 409–444. (page 22.)

MATLAB (2009). *Matlab, version 7.8.0.347 (R2009a)*. The MathWorks Inc., Natick, Massachusetts. (pages 99, 111, and 128.)

Mayraz, G. and Hinton, G. (2002). Recognizing handwritten digits using hierarchical products of experts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **24**(2), 189 – 197. (page 113.)

McKay, M., Beckman, R., and Conover, W. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**(2), 239–245. (page 128.)

Meekings, K. N., Leipzig, J., Bushman, F. D., Taylor, G. P., and Bangham, C. R. M. (2008). HTLV-1 integration into transcriptionally active genomic regions is associated with proviral expression and with HAM/TSP. *PLoS Pathog*, **4**(3), e1000027. (page 51.)

Meinshausen, N. (2007). Relaxed lasso. *Comput Stat Data An*, **52**(1), 374–393. (page 43.)

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *J R Stat Soc B*, **72**(4), 417–473. (pages 25, 27, 28, 31, 35, 36, 38, 39, 40, 48, 63, and 116.)

Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. *J Am Stat Assoc*, **44**(247), 335–341. (page 13.)

Moles, C. G., Mendes, P., and Banga, J. R. (2003). Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res*, **13**(11), 2467–74. (page 104.)

Müller, P., Erkanli, A., and West, M. (1996). Bayesian curve fitting using multivariate normal mixtures. *Biometrika*, **83**(1), 67–79. (page 133.)

Ndao, M., Rainczuk, A., Rioux, M.-C., Spithill, T. W., and Ward, J. (2010). Is SELDI-TOF a valid tool for diagnostic biomarkers? *Trends in parasitology*. (page 52.)

Neal, R. M. (1999). Regression and classification using gaussian process priors. *Bayesian statistics 6*, pages 475–501. (page 118.)

Ochiai, A. (1957). Zoogeographical studies on the soleoid fishes found in japan and its neigbouring regions. *Bull. Jpn. Soc. Sci. Fish*, **22**, 526–530. (page 31.)

Ochs, M. F. (2010). Knowledge-based data analysis comes of age. *Brief Bioinformatics*, **11**(1), 30–9. (page 9.)

Opgen-Rhein, R. and Strimmer, K. (2006a). Inferring gene dependency networks from genomic longitudinal data: a functional data approach. *REVSTAT*, **4**(1), 53–65. (page 80.)

Opgen-Rhein, R. and Strimmer, K. (2006b). Using regularized dynamic correlation to infer gene dependency networks from time-series microarray data. In *The 4th TICSP Workshop on Computational Systems Biology*, pages 73–76. (page 80.)

Opgen-Rhein, R. and Strimmer, K. (2007a). From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Systems Biology*, **1**(1), 37. (pages 10, 67, 79, and 80.)

Opgen-Rhein, R. and Strimmer, K. (2007b). Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics*, **8**(Suppl 2), S3. (page 93.)

Pappin, D. J., Hojrup, P., and Bleasby, A. J. (1993). Rapid identification of proteins by peptide-mass fingerprinting. *Curr Biol*, **3**(6), 327–32. (page 62.)

Perkins, T. J., Jaeger, J., Reinitz, J., and Glass, L. (2006). Reverse engineering the gap gene network of drosophila melanogaster. *PLoS Comput Biol*, **2**(5), e51. (page 96.)

Petricoin, E. F., Ornstein, D. K., Paweletz, C. P., Ardekani, A., Hackett, P. S., Hitt, B. A., Velassco, A., Trucco, C., Wiegand, L., Wood, K., Simone, C. B., Levine, P. J., Linehan, W. M., Emmert-Buck, M. R., Steinberg, S. M., Kohn, E. C., and Liotta, L. A. (2002a). Serum proteomic patterns for detection of prostate cancer. *J Natl Cancer Inst*, **94**(20), 1576–8. (page 52.)

Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V. A., Steinberg, S. M., Mills, G. B., Simone, C., Fishman, D. A., Kohn, E. C., and Liotta, L. A. (2002b). Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, **359**(9306), 572–7. (page 52.)

Phillips, R. B., Kondev, J., and Theriot, J. (2009). *Physical biology of the cell*. Garland Science. (page 11.)

Politis, D. N. and Romano, J. (1994). Large sample confidence regions based on subsamples under minimal assumptions. *The Annals of Statistics*, **22**(4), 2031–2050. (page 18.)

Politis, D. N., Romano, J. P., and Wolf, M. (1999). *Subsampling*. Springer. (page 18.)

Politis, D. N., Romano, J., and Wolf, M. (2001). On the asymptotic theory of subsampling. *Stat Sinica*, **11**(4), 1105–1124. (page 18.)

Pollack, A. (2004). New cancer test stirs hope and concern. *The New York Times*. (page 8.)

Poyton, A., Varziri, M., McAuley, K., McLellan, P., and Ramsay, J. (2006). Parameter estimation in continuous-time dynamic models using principal differential analysis. *Comput Chem Eng*, **30**(4), 698–708. (page 98.)

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, third edition. (pages 32, 96, and 97.)

Pusztai, L., Gregory, B. W., Baggerly, K. A., Peng, B., Koomen, J., Kuerer, H. M., Esteva, F. J., Symmans, W. F., Wagner, P., Hortobagyi, G. N., Laronga, C., Semmes, O. J., Wright, G. L., Drake, R. R., and Vlahou, A. (2004). Pharmacoproteomic analysis of prechemotherapy and postchemotherapy plasma samples from patients receiving neoadjuvant or adjuvant chemotherapy for breast carcinoma. *Cancer*, **100**(9), 1814–22. (page 52.)

Quach, M., Brunel, N., and d'Alché Buc, F. (2007). Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, **23**(23), 3209–16. (pages 95 and 96.)

Quenouille, M. (1949). Approximate tests of correlation in time-series. *Journal of the Royal Statistical Society. Series B (Methodological)*, **11**(1), 68–84. (page 17.)

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: a generalized smoothing approach. *J Roy Stat Soc B*, **69**, 741–770. (page 98.)

Rasmussen, C. (2006). The minimize function. URL: `http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/`. Last accessed: 10/10/2010. (page 71.)

Rasmussen, C. E. (2003). Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. *Bayesian statistics 7*, pages 651–660. (page 128.)

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press. (pages 70, 72, 86, 108, and 109.)

Ratmann, O., Andrieu, C., Wiuf, C., and Richardson, S. (2009). Model criticism based on likelihood-free inference, with an application to protein network evolution. *Proc Natl Acad Sci USA*, **106**(26), 10576–81. (pages 21 and 133.)

Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer-Verlag. (page 13.)

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 139–40. (page 26.)

Rubin, D. (1981a). Estimation in parallel randomized experiments. *Journal of Educational Statistics*, **6**(4), 377–401. (page 20.)

Rubin, D. (1984). Bayesianly justifiable and relevant frequency calculations for the applies statistician. *The Annals of Statistics*, **12**(4), 1151–1172. (page 20.)

Rubin, D. B. (1981b). The Bayesian Bootstrap. *Ann Stat*, **9**(1), 130–134. (pages 14, 18, 19, 20, and 77.)

Rudge, P., Ali, A., and Cruickshank, J. K. (1991). Multiple sclerosis, tropical spastic paraparesis and HTLV-1 infection in Afro-Caribbean patients in the United Kingdom. *J Neurol Neurosurg Psychiatr*, **54**(8), 689–94. (page 63.)

Runarsson, T. P. and Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, **4**(3), 284–294. (page 104.)

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**(19), 2507–17. (pages 10, 27, and 32.)

Saeys, Y., Abeel, T., and Van de Peer, Y. (2008). Robust feature selection using ensemble feature selection techniques. In W. Daelemans, B. Goethals, and K. Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5212 of *Lecture Notes in Computer Science*, pages 313–325. Springer Berlin / Heidelberg. (page 31.)

Saltelli, A., Tarantola, S., and Campolongo, F. (2000). Sensitivity analysis as an ingredient of modeling. *Stat Sci*, **15**(4), 377–395. (page 128.)

Sato, S., Arita, M., Soga, T., Nishioka, T., and Tomita, M. (2008). Time-resolved metabolomics reveals metabolic modulation in rice foliage. *BMC Syst Biol*, **2**, 51. (page 65.)

Sauve, A. C. and Speed, T. P. (2004). Normalization, baseline correction and alignment of high-throughput mass spectrometry data. In *Proceedings of the Genomic Signal Processing and Statistics workshop*. (page 52.)

Schäfer, J. and Strimmer, K. (2005). An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, **21**(6), 754–764. (pages 80 and 81.)

Schäfer, J., Opgen-Rhein, R., and Strimmer, K. (2006). Reverse engineering genetic networks using the genenet package. *R News*, **6**(5), 50–54. (pages 11, 78, 79, 86, 93, and 115.)

Schlitt, T. and Brazma, A. (2007). Current approaches to gene regulatory network modelling. *BMC Bioinformatics*, **8 Suppl 6**, S9. (page 78.)

Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D., and Müller, G. (2002). Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat Biotechnol*, **20**(4), 370–5. (pages 11 and 95.)

Shaffer, J. (1995). Multiple hypothesis testing. *Annual Review of Psychology*. (page 10.)

Shi, L., Tong, W., Fang, H., Scherf, U., Han, J., Puri, R. K., Frueh, F. W., Goodsaid, F. M., Guo, L., Su, Z., Han, T., Fuscoe, J. C., Xu, Z. A., Patterson, T. A., Hong, H., Xie, Q., Perkins, R. G., Chen, J. J., and Casciano, D. A. (2005). Cross-platform comparability of microarray technology: intra-platform consistency and appropriate data analysis procedures are essential. *BMC Bioinformatics*, **6 Suppl 2**, S12. (page 90.)

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, **104**(6), 1760–1765. (pages 118 and 133.)

Smit, S., van Breemen, M. J., Hoefsloot, H. C. J., Smilde, A. K., Aerts, J. M. F. G., and de Koster, C. G. (2007). Assessing the statistical validity of proteomics based biomarkers. *Anal Chim Acta*, **592**(2), 210–7. (page 28.)

Smith, S. M., Fulton, D. C., Chia, T., Thorneycroft, D., Chapple, A., Dunstan, H., Hylton, C., Zeeman, S. C., and Smith, A. M. (2004). Diurnal changes in the transcriptome encoding enzymes of starch metabolism provide evidence for both transcriptional and posttranscriptional regulation of starch metabolism in arabidopsis leaves. *Plant Physiol*, **136**, 2687–2699. (pages 67, 79, and 85.)

Solak, E., Murray-Smith, R., Solak, E., Leithead, W., Rasmussen, C., and Leith, D. (2003). Derivative observations in Gaussian Process models of dynamic systems. In *Advances in Neural Information Precessing Systems 15*, pages 1033–1040. MIT Press. (page 109.)

Somol, P. and Novovičová, J. (2010). Evaluating the stability of feature selectors that optimize feature subset cardinality. *Structural, Syntactic, and Statistical Pattern Recognition*, **5342**, 956–966. (page 30.)

Song, C., Phenix, H., Abedi, V., Scott, M., Ingalls, B. P., Kaern, M., and Perkins, T. J. (2010). Estimating the stochastic bifurcation structure of cellular networks. *PLoS Comput Biol*, **6**(3), e1000699. (page 131.)

Sorace, J. M. and Zhan, M. (2003). A data review and re-assessment of ovarian cancer serum proteomic profiling. *BMC Bioinformatics*, **4**, 24. (page 52.)

Sørensen, L. R. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter*, **5**(4), 1–34. (page 31.)

Stegle, O., Denby, K. J., Cooke, E. J., Wild, D. L., Ghahramani, Z., and Borgwardt, K. M. (2010). A robust Bayesian two-sample test for detecting intervals of differential gene expression in microarray time series. *J Comput Biol*, **17**(3), 355–67. (pages 26 and 67.)

Stein, M. L. (1999). *Interpolation of Spatial Data : Some Theory for Kriging (Springer Series in Statistics)*. Springer. (page 86.)

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, **36**(2), 111–147. (page 28.)

Storey, J. D. (2002). A direct approach to false discovery rates. *J Roy Stat Soc B*, **64**(3), 479–498. (page 54.)

Storey, J. D., Xiao, W., Leek, J. T., Tompkins, R. G., and Davis, R. W. (2005). Significance analysis of time course microarray experiments. *Proc Natl Acad Sci USA*, **102**(36), 12837–42. (pages 82 and 84.)

Strimmer, K. (2008). fdrtool: a versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics*, **24**(12), 1461–2. (page 81.)

Swameye, I., Müller, T. G., Timmer, J., Sandra, O., and Klingmüller, U. (2003). Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modeling. *Proc Natl Acad Sci USA*, **100**(3), 1028–33. (pages 67, 95, 103, and 104.)

Tanimoto, T. (1960). IBM Type 704 Medical Diagnosis Program. *IRE Transactions on Medical Electronics*, **ME-7**(4), 280 – 283. (page 30.)

Taylor, H., Liepe, J., Bugeon, L., Huvet, M., Kirk, P. D. W., Shia, A., Brown, S., Lamb, J. R., Stumpf, M. P., and Dallman, M. J. (2010). In-vivo single cell tracking in zebrafish and statistical modelling reveal heterogeneity in recruitment and chemotactic behaviour in leukocytes. *Submitted*. (page 75.)

Teschendorff, A. E., Wang, Y., Barbosa-Morais, N. L., Brenton, J. D., and Caldas, C. (2005). A variational Bayesian mixture modelling framework for cluster analysis of gene-expression data. *Bioinformatics*, **21**(13), 3025–33. (page 133.)

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J Roy Stat Soc B Met*, **58**(1), 267–288. (pages 32 and 33.)

Tibshirani, R. and Wasserman, L. (2006). Correlation-sharing for detection of differential gene expression. *arXiv*, **math.ST**. (page 11.)

Tibshirani, R., Hastie, T., Narasimhan, B., Soltys, S., Shi, G., Koong, A., and Le, Q.-T. (2004). Sample classification from protein mass spectrometry, by 'peak probability contrasts'. *Bioinformatics*, **20**(17), 3034–44. (page 53.)

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. H. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *J R Soc Interface*, **6**(31), 187–202. (pages 96, 97, 98, 117, and 129.)

Toulza, F., Heaps, A., Tanaka, Y., Taylor, G. P., and Bangham, C. R. M. (2008). High frequency of CD4+FoxP3+ cells in HTLV-1 infection: inverse correlation with HTLV-1-specific CTL response. *Blood*, **111**(10), 5047–53. (page 51.)

Tukey, J. W. (1958). Bias and confidence in not-quite large samples. *The Annals of Mathematical Statistics*, **29**(2), 614–623. (page 17.)

Turney, P. (1995). Technical note: Bias and the quantification of stability. *Machine Learning*, **20**(1-2), 23–33. (page 27.)

Tyson, J. J., Chen, K. C., and Novak, B. (2003). Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. *Curr Opin Cell Biol*, **15**(2), 221–31. (page 95.)

van de Vijver, M. J., He, Y. D., van't Veer, L. J., Dai, H., Hart, A. A. M., Voskuil, D. W., Schreiber, G. J., Peterse, J. L., Roberts, C., Marton, M. J., Parrish, M., Atsma, D., Witteveen, A., Glas, A., Delahaye, L., van der Velde, T., Bartelink, H., Rodenhuis, S., Rutgers, E. T., Friend, S. H., and Bernards, R. (2002). A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, **347**(25), 1999–2009. (page 26.)

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**(6871), 530–6. (page 26.)

Varah, J. (1982). A spline least-squares method for numerical parameter-estimation in differential-equations. *Siam J Sci Stat Comp*, **3**(1), 28–46. (page 97.)

Vine, A. M., Heaps, A. G., Kaftantzi, L., Mosley, A., Asquith, B., Witkover, A., Thompson, G., Saito, M., Goon, P. K. C., Carr, L., Martinez-Murillo, F., Taylor, G. P., and Bangham, C. R. M. (2004). The role of CTLs in persistent viral infection: cytolytic gene expression in CD8+ lymphocytes distinguishes between individuals with a high or low proviral load of human T cell lymphotropic virus type 1. *J Immunol*, **173**(8), 5121–9. (page 26.)

Voit, E. O. and Almeida, J. (2004). Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, **20**(11), 1670–81. (page 97.)

Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, **118**, 558–560. (page 98.)

von Mises, R. (1964). *Mathematical Theory of Probability and Statistics*. Academic Press, New York. H. Geiringer, ed. (page 74.)

Šidàk, Z. (1968). On multivariate normal probabilities of rectangles: Their dependence on correlations. *The Annals of Mathematical Statistics*, **39**(5), 1425–1434. (page 10.)

Werhli, A. V., Grzegorczyk, M., and Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, **22**(20), 2523–31. (pages 94 and 117.)

West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian statistics 7*, pages 733–742. (page 9.)

Wichert, S., Fokianos, K., and Strimmer, K. (2004). Identifying periodically expressed transcripts in microarray time series data. *Bioinformatics*, **20**(1), 5–20. (page 79.)

Wilkinson, D. J. (2006). *Stochastic modelling for systems biology*. Chapman and Hal/CRCl. (pages 11, 98, and 129.)

Wilkinson, M. M., Busuttil, A., Hayward, C., Brock, D. J., Dorin, J. R., and Van Heyningen, V. (1988). Expression pattern of two related cystic fibrosis-associated calcium-binding proteins in normal and abnormal tissues. *J Cell Sci*, **91 (Pt 2)**, 221–30. (page 62.)

Wilks, S. S. (1963). *Mathematical Statistics*. Wiley. (page 19.)

Wise, R., Caldo, R., Hong, L., Shen, L., Cannon, E., and Dickerson, J. (2007). BarleyBase/PLEXdb: a unified expression profiling database for plants and plant pathogens. In *Methods in Molecular Biology*, volume 406, pages 347–363. Humana Press. (page 119.)

Xie, J. and Yi, Q. (2003). β2-microglobulin as a potential initiator of inflammatory responses. *Trends Immunol*, **24**(5), 228–9; author reply 229–30. (pages 62 and 116.)

Yu, L., Ding, C., and Loscalzo, S. (2008). Stable feature selection via dense feature groups. In *Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining (KDD08)*. (page 28.)

Yuan, M. (2006). Flexible temporal expression profile modelling using the Gaussian process. *Comput. Stat. Data Anal.*, **51**(3), 1754–1764. (page 67.)

Zhang, M., Yao, C., Guo, Z., Zou, J., Zhang, L., Xiao, H., Wang, D., Yang, D., Gong, X., Zhu, J., Li, Y., and Li, X. (2008). Apparently low reproducibility of true differential expression discoveries in microarray studies. *Bioinformatics*, **24**(18), 2057–63. (page 8.)

Zhang, M., Zhang, L., Zou, J., Yao, C., Xiao, H., Liu, Q., Wang, J., Wang, D., Wang, C., and Guo, Z. (2009). Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes. *Bioinformatics*, **25**(13), 1662–8. (page 8.)

Znamenkiy, P. (2006). JAK STAT pathway (public domain). URL: `http://upload.wikimedia.org/wikipedia/commons/6/6e/Jakstat_pathway.svg`. Last accessed: 10/10/2010. (page 103.)

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J Roy Stat Soc B*, **67**, 301–320. (pages 28, 33, and 34.)

Zuber, V. and Strimmer, K. (2009). Gene ranking and biomarker discovery under correlation. *Bioinformatics*. (pages 11 and 57.)

Zucknick, M., Richardson, S., and Stronach, E. A. (2008). Comparing the characteristics of gene expression profiles derived by univariate and multivariate classification methods. *Statistical applications in genetics and molecular biology*, **7**(1), 7. (pages 27, 29, and 31.)